最適交通網構成手法に関する基礎的研究(Dissertation_全文)

AUTHOR(S):

森津, 秀夫

最適交通網構成手法に関する基礎的研究

昭和59年2月

森　津　秀　夫

序

道路や鉄道などはネットワークを形成し、その機能をはたしている。これらを対象とする交通網計画において一般に使用されてきた方法は、いくつかの代替案を比較し、もっとも好ましい交通網案を選択するものである。けれども、この方法は、代替案の作成がもっぱら計画者の主観によるために最適な計画案が含まれない可能性があることや、ネットワークが複雑なときに代替案作成作業が困難であるという欠点を持つ。そこで、客観的な基準を使い、最適化手法によって交通網計画を行うことが必要になり、そのために最適交通網構成手法が研究されてきた。

しかし、これまでの研究は単純な問題の解法を扱い、簡単な例題を解いている程度のものがほとんどであった。実際的な交通網計画に適用するには、なお多くの課題が残されていたといえる。そこで、本研究では最適交通網構成手法を実用的な水準に近づけることを目的とし、解法と応用に関する基礎的な考察を行う。

本研究では、まず基本的な最適交通網構成手法を対象に、改良した厳密解法と近似解法を提案する。つぎに、基本的な問題の拡張に際して必要な解法の修正について考察する。すなわち、制約条件が複数の問題、目的関数が複数の問題、段階建設問題、不確実な需要交通量を用いる問題の解法を提案する。さらに、地域交通網計画問題、多目的を扱う道路網計画問題、道路網の対災害信頼性の最適化問題、配分対象道路網作成問題、バス路線網計画問題に最適交通網構成手法を適用する。これにより、最適交通網構成問題としての定式化と解法の改良の実際を示し、最適交通網構成手法の実用性を明らかにする。

本研究の遂行にあたっては神戸大学工学部枝村俊郎教授に、そして、論文の作成に際しては京都大学工学部佐佐木綱教授に御指導と御鞭撻を賜った。ここに、心から感謝の意を表する次第である。また、本研究の遂行に御協力と御助力をいただいた神戸大学工学部土木工学科交通工学研究室の諸氏に感謝の意を表したい。

昭和59年2月

森　津　秀　夫

最適交通網構成手法に関する基礎的研究

目　　次

第１章　序　　論

１．１　本研究の目的

道路や鉄道などはネットワークを形成し、その機能をはたしている。そのため、これらの施設の新設や改良などの計画は、ネットワークをどのようにするかという観点から行なわれる。これが交通網計画である。このとき、一般に使用されてきた方法はいくつかの代替案を比較し、もっとも好ましい交通網案を選択するものである。しかし、この方法では代替案の作成はもっぱら計画者の主観によるため、最適な計画案が代替案に含まれないことが起り得る。ネットワークが複雑な場合には、代替案の作成そのものが困難でさえある。そこで、客観的な基準によって最適な交通網案をつくる方法を開発しようという試みがなされてきた。すなわち、最適化手法により交通網計画を行なおうとするのであり、これが最適交通網構成手法である。だが、これまでの研究では簡単な例題が解ける程度でしかなかった。実際的な交通網計画に適用するには、多くの課題が残されていた。

本研究の目的は、最適交通網構成手法を実用的な水準に近づけるための基礎的な考察を行うことである。そのために必要なことは、次のようにまとめられる。

① 最適交通網構成問題の最適解を効率よく求めるための解法の作成。

② 最適交通網構成問題の近似解を短い計算時間で求められる実用的な解法の作成。

③ 現実的な交通網計画を最適交通網構成問題として扱うための定式化法と、解法の拡張に関する考察。

これらを考慮し、本研究では基本的な最適交通網構成問題を対象に、改良した厳密解法と近似解法を提案する。そして、その基本的な問題の拡張に際して必要な解法の修正について考察する。さらに、いろいろな交通網計画問題に最適交通網構成手法を適用し、定式化と解法の改良の実際を示す。

１．２　本研究の内容

この研究は７章によって構成している。第２章は最適交通網構成問題の概略をまとめたものである。第３章と第４章は、基本的な問題を対象とした解法の考察である。第５章と第６章は、現実的な交通網計画を扱うためのものであり、基本的な問題の拡張と応用例を示す。そして、第７章は、本研究の結論である。以下に、第２章以降の内容を簡単にまとめる。

第２章では、本研究の対象とする最適交通網構成問題がどのようなものであるかを明確にする。最初に交通網計画で一般に使われてきた代替案を比較する方法の欠点を指摘し、最適化手法の導入の必要性を示す。つぎに、交通網計画で使われる決定変数や目的関数、制約条件について検討し、ここで扱う最適交通網構成問題を明示する。そして、最適交通網構成問題の特徴を有するもっとも簡単な問題を、最適交通網構成問題の基本問題として定式化する。さらに、最適交通網構成問題の解はどのようにして求められるか、解法の問題点は何か、解法のどのような研究がなされているかを述べる。最後に、現実の交通網計画に適用しようとするとき、基本問題をどう拡張すればよいかを考察する。

第３章では、基本問題を対象に厳密解法を考察する。まず従来のアルゴリズムを検討し、その結果からアルゴリズムの改良方法を考える。そして、実行可能解まで一度に分枝する実行可能化分枝を用いるアルゴリズムを提案する。さらに、最適解は連結網であることを利用し、解集合に属する解がすべて連結網になるように分枝する連結網化分枝を用いるアルゴリズムを提案する。最後に例題を解き、提案した実行可能化分枝と連結網化分枝の効果を調べる。

第４章では、基本問題を対象に近似解法を考察する。最初に従来の近似解法を検討し、その結果からどのように新しい近似解法を作ればよいかを考える。そして、厳密解法を繰り返し適用してリンクの段階的削減を行うものと、backward 法の解の近傍において厳密解法を使って局所最適化を行うものとの２種類の近似解法を提案する。さらに、大規模な最適交通網構成問題を解くのに適した近似解法を提案する。すなわち、簡単なリンク評価値を使って forward 法、または backward 法の手順で解を求めるものであり、これを応用した解法も提案する。最後に、例題を解いて必要な計算時間と解の精度を調べ、提案した近似解法がこれまでの解法よりも優れていることを確かめる。

第５章では、最適交通網構成問題を拡張する場合の基本的なものを取り上げ、どのような解法の修正が必要かを考察する。複数の制約条件を持つ最適交通網構成問題に関しては、制約条件の種類によってどのように解けばよいかを検討する。そして、多次元ナップザック問題と同じ制約条件を持つ場合を対象とする解法を提案する。複数の目的関数を持つ多目的最適交通網構成問題に関しては、一般の多目的計画問題との相違を明らかにする。そのうえで、適用が可能な目的関数のスカラ手法を用いる解法を提案する。つぎに、交通網を長期間にわたって段階的に建設する問題を最適交通網構成問題として定式化し、建設途中の需要交通量をも考慮する場合の解法を提案する。最後に、需要交通量が不確実なとき、その不確実さを考えなければ損失を被る場合があることを示す。そして、需要交通量の不確実さを考慮した解法を提案する。

第６章では、いろいろな交通網計画問題への最適交通網構成手法の応用について述べる。現実的な交通網計画問題を解いて最適交通網構成問題の実用性を示すとともに、これまで交通網計画問題として扱われていなかった問題へも適用し、応用の可能性が大きいことを示す。最初に、鉄道、新交通システム、バスの３種類の大量輸送機関を適切に組み合わせる地域交通網計画問題への応用について述べる。つぎに、騒音と排出ガスの影響をも考慮して道路網を計画する問題を多目的最適交通網構成問題として解く。道路網については、災害に対する信頼性を最適化する問題も考える。そして、交通配分を行うとき、配分対象道路網はどのようなものであればよいかを検討し、客観的な基準で道路網案を作成する手法を提案する。最後に、バス路線網計画問題への適用を行う。これはリンクの組み合わせを決めるのでなく、路線系統の最適な組み合わせを求める点が特徴である。

第７章は本研究の結論であり、第２章～第６章で得られた結果をまとめる。

第２章　最適交通網構成問題に関する基礎的考察

２.１　概　　説

交通網を計画するとき、何らかの意味において最適な計画の作成を目指している。そこで、交通網計画問題を最適化問題として定式化したのが最適交通網構成問題である。この章では、本研究で対象とする最適交通網構成問題に関する基礎的な考察を行う。最初に交通網計画問題への最適化手法の導入について述べ、最適交通網構成問題を定式化する。つぎに最適交通網構成問題の解法について述べ、最後に基本的な問題の拡張に関して考察する。

２.２では、交通網計画で一般に使われている手法の欠点を指摘し、最適化手法の導入の必要性を述べる。

２.３では、交通網計画で使われる目的関数や制約条件について述べ、もっとも単純な最適交通網構成問題を定式化する。

２.４では、最適交通網構成問題の解法に関する考察を行う。

２.５では、現実の交通網計画を対象とするとき、基本的な問題をどのように拡張しなければならないかを述べる。

２.６では、この章で考察したことをまとめる。

２.２　交通網計画問題への最適化手法の導入

道路や鉄道、バスなどはネットワークを形成し、交通需要に対する機能を果たしている。そのため、交通網の一部分だけに手を加えても、その影響が交通網全体に及ぶことが多い。そこで、交通網の構成要素に関する計画を行うとき、単に個々の部分の計画として考えるのでなく、交通網全体の評価を行う交通網計画として扱うことが必要である。ここでは、交通網計画で一般に使われている手法の欠点を指摘し、最適化手法を導入する必要性を述べる。

交通網の構成要素は、道路区間のような交通施設の区間と、交差点のような区間の結節点である。交通網に関する計画は、このどちらを対象とするものかによって区別される。一般に交通網計画といわれるのは、交通施設の区間を対象とするものである。しかも、計画の対象となる区間が多数あることが多い。ここで扱う交通網計画も、このようなものである。すなわち、ノードとリンクからなるネットワークで、リンクの建設や改良を検討する計画である。この場合、ノードは交通の発生、集中地点としての意味をもつが、改良などの計画は考えない。

交通網計画案を策定するとき、一般に使われる手法は少数の代替案を比較、検討し、ひとつの案を選択するものである。まず比較の対象となる代替案を作成し、それぞれの代替案の評価指標を計算する。そして、それをもとに代替案の優劣を判定する。評価指標の計算以外は手作業である。場合によっては、これらの作業を試行錯誤により繰り返すこともある。代替案の作成は計画者の主観によることになる。そのため、この手法は次に示す欠点をもつ。

①　作成した代替案よりも良い交通網案が存在する可能性がある。

②　代替案になり得る交通網が多数あるとき、少数の代替案を決めることが困難である。

この手法で決定した交通網計画案は、最良の計画であるとの保証はなく、計画としての説得力が弱い。そして、たとえば新たな地域に交通網をつくるような場合には多数の交通網案が考えら

れ、少数の代替案を選ぶことが容易でないのである。

これらの欠点を除くには、交通網計画案の策定を最適化問題として整理し、数理計画手法を適用しなければならない。すなわち、客観的な基準を設け、それを最適にする交通網を求める問題を考えなければならない。そして、交通網案は問題の内部で作成されるようにする。このような問題を解けば、ある基準においては最適であることが保証された交通網案が得られる。そして、問題を解く段階で交通網案が作成されるため、代替案の選択に困ることもない。

最適化手法を用いた場合、従来の方法よりも多くの交通網案を調べることになる。交通網の評価方法は場合によって異なるが、交通配分か最短経路探索を必要とすることがほとんどである。交通網が非常に簡単でない限り、これらを手作業で行うことはできない。したがって、多数の交通網を調べて最適化問題を解くには、コンピューターを用いることが前提となる。

2.3 最適交通網構成問題の定式化

最適化手法によって交通網案を求めるには、まず問題を定式化しなければならない。しかし、単に交通網計画といっても対象はさまざまである。あらゆる交通網計画問題をひとつの具体的な問題で表わすことはできない。そこで、ここでは交通網計画において使われる目的関数や制約条件について述べる。そして、この研究で対象とする交通網計画の前提条件と、どのような立場で交通網を計画するかを明確にする。そのうえで、最適交通網構成問題としての特徴を備えた、もっとも単純な問題を示す。

ここで対象とする交通網計画問題は、制約条件を満たし、最適な目的関数値を与えるネットワークを求めるものである。目的関数は最小化するものがひとつだけであるとすると、交通網の最適化問題は次のように表わせる。

問題2.1

$$\min \quad f(\boldsymbol{x}) \tag{2.1}$$

$$\text{s.t.} \quad \boldsymbol{g}(\boldsymbol{x}) \gtreqless \boldsymbol{0} \tag{2.2}$$

ここに、f：目的関数

$\boldsymbol{g}$：制約条件

$\boldsymbol{x}$：ネットワークの状態を表わす変数

問題2.1の定式化は数理計画問題の一般的な表現と変わらない。これを交通網の最適化問題として特徴づけるのは、決定変数がネットワークを表わすことである。そして、最短距離のようにネットワークが定まって決まるものを目的関数か制約式に含むことにおいて、他の数理計画問題と異なる。このような問題を最適ネットワーク問題、ネットワーク最適化問題あるいは最適ネットワーク構成問題などとよんでいる。ここでは、対象が交通網であることを明確にし、最適交通網構成問題とよぶことにする。

問題2.1において、決定変数と目的関数、制約条件を定めれば、具体的な最適交通網構成問題を定式化できる。そこで、これらについて述べてゆく。しかし、すでに西村・日野[1]、Steenbrink[2]などによってまとめられている。そこで、ここでは詳細に論じることはしない。

最適交通網構成問題の決定変数は、全体でネットワークの状態を表わすものでなければならな

い。しかも、計画を実施するときに操作できなければ決定変数としての意味を持たない。ここで対象としている交通網計画は、リンクの建設や改良を検討するものである。したがって、各リンクの計画を反映する決定変数を選ぶことが必要である。たとえば、リンクを建設するかしないか、あるいは道路網計画における車線数や道路幅員などである。

リンクの計画の実施を表わす決定変数に加え、リンク交通量なども決定変数に用いられることがある。これは、交通網の利用者の行動も操作可能であるとし、利用法をも含めて計画するものである。交通配分でよく議論されるシステム最適と利用者最適のうち、システム最適を図ることになる。この場合、利用者の経路を規定する有効な手段の存在が前提である。交通網の利用法に関するものを決定変数にしないときは、利用者がそれぞれ自分に最適な利用法をとると仮定することになる。

決定変数の選択によっては、問題の性格が大きく異なる。それは、決定変数が連続変数か、離散変数かである。道路幅員や交通容量、輸送能力、交通量などは連続変数として扱える。一方、車線数や建設の可否などは離散変数である。決定変数が連続変数だけのときは、問題は線形計画問題か非線形計画問題になる。離散変数だけのときは整数計画問題で、両者が混在するときは混合整数計画問題である。実際の交通網計画では、決定変数の値が限られることが多い。そこで、連続変数として扱えるものでも、選択肢のどれを選ぶかを決定変数にする方が現実的である。そうすれば、ほとんどの交通網計画問題は整数計画問題として考えてよい。

目的関数に何を用いるかは、どのような立場で交通網計画を行うか、そして交通網をどう評価するかによる。一般に使用が考慮されるのは、費用、便益、利用者の余剰である。[2)] これらを組み合わせた、たとえば費用便益差や費用便益比も使われる。また、効用関数で表わされることもある。具体的な指標には、総走行距離、総走行時間、建設費、騒音や排出ガスによる外部不経済などがある。

制約条件に用いられるものは、目的関数に使用されるものとほぼ同じである。計画に際して考慮しなければならないもののうち、ある項目だけを目的関数とし、他を制約条件にするのがよく使われる方法である。制約条件は最適解において必ず満たされるが、制約値との余裕は考慮されない。目的関数にした場合は一定値以下、あるいは一定値以上になることは保証されないが、可能な限り小さくあるいは大きくされる。これらの点を考えて、目的関数と制約条件に分けるのである。使用する解法の制約のため、本来は目的関数にすべきものを制約条件にすることもある。

総走行距離と総走行時間は交通網の機能を表わすもっとも重要な指標である。ほとんどすべての交通網計画問題で、少なくともどちらか一方が目的関数か制約条件に使われる。総走行距離や総走行時間は交通配分問題の解から得られる。利用者の経路選択を決定変数にしないとき、交通配分方法が決まっていれば、これらは交通網に対して一意に定まる。そのため、これらを交通網の関数と考えることができる。交通配分結果から求める騒音や排出ガスについても同様であり、これらも交通網の関数と考えられる。このような交通網によって値の定まるものを目的関数か制約条件に含むことが最適交通網構成問題の特徴である。

総走行距離を求めるには交通配分問題を解かなくてはならない。したがって、総走行距離を使用するときには、最適交通網構成問題は交通配分問題を内在することになる。交通配分が本来の問題の目的関数を最適化するようになされるときは、交通配分問題は最適交通網構成問題の欠か

せない一部分である。そうでないときは、交通配分問題は補助問題と見なせる。いずれにしても、問題を完全に表現するには交通配分問題も定式化に含めなければならない。そうすると、交通流の保存則を表わす式も制約条件に加えることになる。しかし、最短経路探索を用いた交通配分手法を使うときには、交通流の保存則の式を明示的に取り扱う必要のないことが多い。そこで、とくに必要のない限り、最短距離や配分交通量などは交通網の関数として扱い、これらを求めるための式は定式化に含めない。

交通網の変化が交通需要に及ぼす影響も、触れておかなければならない問題である。一般に交通網は土地利用などを前提とし、与えられた交通需要に対して計画される。しかし、交通網の変化は土地利用の変化を促し、その結果、交通需要も変化する。交通網計画案は変化した後の交通需要に対して最適であることが望ましい。すなわち、交通需要の変化を見込み、問題に取り入れておくことが必要である。だが交通網の変化による交通需要の変化を正確に予測する方法は確立されていない。そこで、従来の交通網計画の手法と同様に交通需要はあらかじめ与えられるものとし、交通網の変化による交通需要の変化は考えない。

決定変数、目的関数、制約条件に関しては、以上のとおりである。これらを使って最適交通網構成問題を定式化するとき、ここでは交通網を建設する側に対するものを制約条件とし、利用者の利便性を表わすものを目的関数に選ぶようにする。これは、可能な資源を有効に使って利用者に便利な交通網を計画する立場をとることを意味する。そして、交通網の利用者はそれぞれに最適な経路を選ぶものとし、利用者の経路を規定する決定変数は設けない。

対象とする最適交通網構成問題に関して述べてきたことをまとめると、次のようになる。

① 決定変数はリンクの計画の実施を表わす離散変数とする。そのため、対象とする最適交通網構成問題は整数計画問題となる。

② 目的関数は利用者の利便性を表わすものとする。

③ 制約条件は交通網を建設する側に対するものとする。

④ 利用者はそれぞれに最適な経路を選ぶものとする。

⑤ 交通網の変化による交通需要の変化は考えない。

この研究で扱う最適交通網構成問題は、原則として、これらを前提としたものとする。しかし、これらは前提条件として必須ということではなく、むしろ交通網計画をどのような観点で行うかを明らかにするものである。単に交通網を計画するというだけでは対象範囲が広く、共通して使える解法を考えることが困難である。そこで、これまでの最適交通網構成問題の研究でもよく使用される問題を含み、交通網計画で一般的な範囲に限定する。さまざまな交通網計画問題を取り上げてゆけば、これらの範囲では扱えないものも生じる。だがそのような場合でも、ここで対象とする問題の考察を基礎にできる。同じ解法が使えるか、少しの手直しでよい場合が多いであろう。

このような最適交通網構成問題の解法を考えるとき、もっとも基本的な問題を最初に対象とすべきである。決定変数のもっとも簡単な場合は、リンクを建設するか否かである。目的関数としては総走行距離の最小化が考えられる。そして、需要交通量がすべてのノード間で等しいとすれば、最短距離の和を求めるだけでよい。制約条件には交通網の建設費の予算制約が考えられる。だが、リンクの建設費がリンクの長さに比例するとすれば、これもネットワークの総延長の制約

に置き換えられる。すなわち、最適交通網構成問題としての特徴を有し、もっとも単純な問題は、ネットワーク長の制約下で最短距離の合計を最小にするネットワークを求めるものである。これは、次のように定式化できる。

問題２．２

$$\min \quad Z=\sum_i \sum_j d_{ij}(\boldsymbol{x}) \tag{2.3}$$

$$\text{s.t.} \quad \sum_k l_k x_k \leqq L^c \tag{2.4}$$

$$x_k = 0 \ or \ 1 \tag{2.5}$$

ここに、Z：ノード間の最短距離の合計値

d_{ij}：ノードi j間の最短距離

$\boldsymbol{x}=\{x_1, x_2, \ldots, x_k, \ldots\}$

：ネットワークの状態を表わし、リンクkがネットワークに含まれないとき$x_k=0$、ネットワークに含まれるとき$x_k=1$とする。

l_k：リンクkの長さ

L^c：ネットワークの総延長の制約値

式（２.３）は、ネットワークの関数として得られるノード間の最短距離の合計を最小化する目的関数である。式（２.４）はネットワークの総延長を制約値以下にすることを表わしている。そして、式（２.５）は各リンクの状態はネットワークに含まれるか否かのどちらかであることを表わしている。

問題２.２は Scott、[3] Hoang[4]などの研究でも使われているものである。問題の構造が簡単であるため、解法の検討に適している。しかも、最適交通網構成問題の基本的な特徴を備えている。そのため、この問題に対して開発した解法を他の問題に応用することが可能である。そこで、問題２.２を最適交通網構成問題の基本問題とよぶことにする。

２．４　最適交通網構成問題の解法

最適交通網構成問題は、個々のリンクの計画をもとに最適なネットワークを求めるものである。数理計画問題として見れば、整数計画問題の一種である。とくにリンクの計画の採否を決定変数とした場合は０－１整数計画問題となる。したがって、最適交通網構成問題の解法は０－１整数計画問題や整数計画問題の解法を基礎にできる。

０－１整数計画問題は決定変数に対応するプロジェクトのうち、どれを採択するかという問題である。つまり、最適なプロジェクトの組み合わせを求めるのである。そのため、この問題は組み合わせ問題ともよばれる。変数の値は０か１のどちらかで、その組み合わせが解を表わす。そして、最適解は実行可能解の中で目的関数値が最小あるいは最大のものを探せばよい。そこで、０－１整数計画問題を解くには変数の組み合わせをすべて列挙し、制約条件を満たし、目的関数値を最適にするものを選べばよい。これが陽的列挙法や完全列挙法とよばれる解法である。しかし、これでは計算効率が悪いので、制約条件を満たさないことが明らかな解や最適解になる可能性のない解は列挙しないようにした解法が使われる。これが陰的列挙法である。[5)6)]

最適交通網構成問題も陰的列挙法を用いて解ける。陰的列挙法では解を系統的に列挙するため

に探索木を使い、分枝操作を繰り返す。この分枝方法と解の探索範囲を限定する方法の違いでさまざまな解法がある。これらの解法が通常の０－１整数計画問題と異なるのは、交通網を最適化する問題の性質を利用していることである。たとえば問題２.２を解くとき、あるネットワークからリンクを除けば、目的関数値は変わらないか増加するかであるという性質が使われる。

０－１整数計画問題は、変数が多くなれば解である変数の組み合わせの数が急激に増大し、解を得るのが困難になる。これは、最適交通網構成問題の場合にはさらに深刻な問題である。最適交通網構成問題の解を評価するには最短経路探索や交通配分を行わなくてはならない。変数が多いことは、リンクの多い複雑なネットワークを扱わなければならないことを意味している。そのようなネットワークでは、最短経路探索に要する計算時間は大きい。よって、個々の解を評価するのに必要な計算時間が増大する。したがって、最適交通網構成問題の規模が大きくなったとき、解を求めるための計算時間は極めて大きくなることがわかる。

最適交通網構成問題は、数学的な関心の対象となるだけのものではない。交通網計画と密接に結びついた問題である。そのために、問題の規模が大きくなることは避けられない。そこで、得られる解は必ずしも最適解でなくともよいから、大きな問題を短い計算時間で解ける実用的な解法が研究されている。最適解が必ず求まる解法を厳密解法と呼び、解の厳密性を捨てた解法を近似解法と呼んでいる。

近似解法においても、実際的な交通網計画問題を扱うために計算時間の短縮を図らなければならないことは同じである。そして、得られる解は最適解ではなくてよいとしても、できるだけ良い解であることが望ましい。近似解法の開発に際しては、このふたつが課題である。しかし、どちらをも解決することは容易でない。それぞれの場合に要求される解の精度を満たす範囲で、可能な限り計算時間の短い解法を考えることになる。

以上に述べたことをまとめると、次のようになる。

① 最適交通網構成問題は陰的列挙法を用いて解ける。

② 最適交通網構成問題では解を評価するのに最短経路探索などが必要である。そのため、問題の規模が大きくなったときの計算時間の増加は通常の０－１整数計画問題よりも急激である。

③ 実際的な問題を扱うため、短い計算時間で最適解に近い解を求められる近似解法が考えられている。

④ 近似解法の開発においては、解の精度の向上と計算時間の短縮が課題である。

２.５ 基本問題の拡張に関する考察

最適交通網構成問題の解法の研究には、問題２.２に示した基本問題が使われる。しかし、この基本問題は現実的なものではなく、そのままの形で実際の交通網計画に使われることはほとんどない。応用に際しては、それぞれの交通網計画問題に合わせて定式化し直さなければならない。このとき、基本問題の拡張方法はいくつかに分類できる。ここでは、交通網計画を最適交通網構成手法によって行うとき、基本問題をどのように拡張して応用すればよいかを述べる。

基本問題の拡張は、基本問題の定式化の際の前提条件や仮定を除くことである。２.２で述べた

ものや、暗黙のうちに前提としていたものを除くと、次のような問題の拡張が考えられる。

① 需要交通量の導入

② ネットワーク長制約の建設費制約への変更

③ 固定リンクの導入

④ 制約条件が複数の問題への拡張

⑤ 多目的計画問題への拡張

⑥ 段階建設問題への拡張

⑦ 需要交通量の不確実性の導入

⑧ 0－1整数計画問題から整数計画問題への変更

⑨ 複数の交通機関を扱う問題への拡張

①～③は軽易な拡張で、解法は基本問題とほとんど変わらない。一般に拡張として考えられているのは④～⑨のものである。これらのいくつかについては、森地がまとめている。[7)]

基本問題の制約条件はネットワークの総延長に関するものだけである。この制約式はナップザック問題と同じものであり、実行可能解を求めるときには扱いやすい。さまざまな形の複数の制約式がある問題を解くには、基本問題に対する解法をかなり修正しなくてはならない。

数理計画問題の多くは、ただひとつの目的関数を最適化するものである。しかし、現実の問題では複数の目的関数を考えなければならないことがある。そのために、多目的計画問題の解法が研究されている。これは交通網計画においても同じである。複数の目的関数をもつ最適交通網構成問題を解くことが必要になろう。

基本問題においては、計画した交通網が完成するまでに要する時間は考えていない。交通網の建設には長期間を要し、したがって計画年次はかなり先になるのが普通である。そして建設期間をいくつかに分け、段階的に建設することが行われる。このような場合、単に完成時だけでなく、建設途中の段階での交通網と需要交通量の関係を考慮しなくてはならない。これが交通網の段階建設問題である。

交通網計画を行うとき、用いる需要交通量は将来の予測値である。また交通量はつねに一定ということはなく、絶えず変動している。通常は需要交通量の推定誤差や変動は考えないで、その平均値だけを使っている。しかし、問題によっては、需要交通量の不確実さを考慮しなければ損失を被ることがある。そこで、不確実な需要交通量に対して最適な交通網を求める手法が必要である。

基本問題における決定変数はリンクの建設の可否を表わす0－1変数であった。そのため、基本問題は0－1整数計画問題と見なせた。だが、あるリンクに関する計画が複数あり、そのどれを採用するかあるいは採用しないかを決める場合には、決定変数は0－1でない整数変数となる。そして、問題は整数計画問題の一種になる。この問題は0－1整数計画問題に変換できるが、むしろ解の探索方法を作り直した方がよい。選択肢が増えるので計算方法の効率化がより一層重要である。

多種類の交通機関の交通網を計画するとき、交通機関ごとに計画するよりも全体をひとつの計画とした方が好ましい交通網が得られる。この場合、交通機関別分担率を与えるモデルを最適交

通網構成問題に組み込むことになる。しかし、交通機関別分担率が正確に推定できなければ、最適ではない計画案が作られる恐れがある。したがって、交通機関別分担率モデルが重要である。十分な精度で交通機関別分担率を推定できない場合、複数の交通機関の交通網を同時に決定するメリットが失われるかもしれない。

以上に述べた問題の拡張は単独にあるのでなく、これらを組み合わせた問題を考えなければならない。そのとき、個々の拡張に際して開発した解法を応用すればよいが、新たに解法を検討し直さなくてはならないことも考えられる。だがその場合においても、それぞれの拡張において解法を検討した手順が役立つであろう。

2.6 結　　語

この章では、最適交通網構成問題に関する基礎的な考察を行った。交通網計画への最適化手法の導入の必要性を述べ、最適交通網構成問題を定式化してその解法について考察した。そして、基本的な問題の拡張を検討した。

交通網計画案を策定する従来の方法は少数の代替案を比較するものである。この方法の欠点として示したのは次の2点である。

① 作成した代替案よりも良い交通網案が存在する可能性がある。

② 代替案になり得る交通網が多数あるとき、少数の代替案を決めることが困難である。

これらの欠点を除くには、交通網計画案を最適化手法を用いて策定する必要があることを述べた。

最適交通網構成問題の定式化に際しては、決定変数と目的関数、制約条件について検討した。そして、ここで扱う最適交通網構成問題を明確にした。さらに、交通網の関数として値の定まるものを目的関数か制約条件に含むという特徴を備え、もっとも単純な最適交通網構成問題を定式化した。すなわち、ネットワーク長の制約下で最短距離の合計を最小化するリンクの組み合わせを求める問題である。この問題は、これまでにも解法の研究に多く使われてきた基本的なものである。そこで、ここではこれを最適交通網構成問題の基本問題と定めた。

最適交通網構成問題の解法に関しては、次のことを述べた。

① 最適交通網構成問題は陰的列挙法を用いて解ける。

② 解の評価に最短経路探索などが必要なため、問題の規模が大きくなったときの計算時間の増加は急激である。

③ 実際的な問題を扱うための近似解法が考えられている。

④ 近似解法の開発においては、解の精度の向上と計算時間の短縮が課題である。

基本問題の拡張については、いくつかのパターンを示した。そのうち、解法の大きな変更を伴うものや実際の交通網計画に応用するときに重要なのは、次のものであった。

① 制約条件が複数の問題への拡張

② 多目的計画問題への拡張

③ 段階建設問題への拡張

④ 需要交通量の不確実性の導入

⑤　整数計画問題への拡張

⑥　複数の交通機関を扱う問題への拡張

ここに述べてきたように、最適交通網構成手法に関する研究は、基本問題に対する厳密解法と近似解法の開発が基礎になる。そして、現実的な交通網計画問題へ応用するために対象とする問題を拡張し、解法を検討するのである。

参考文献

1） 西村　昂・日野泰雄：最適ネットワーク構成に関する一考察、土木学会論文報告集、第250号、pp.85～97, 1976年6月.

2） Steenbrink, P.A. : Optimization of Transport Networks, John Wiley & Sons,1974.

3） Scott, A.J. : The optimal network problem : some computational procedures, Transportation Research, Vol. 3, pp. 201～210, 1969.

4） Hoang, H.H. : A computational approach to the selection of an optimal network, Management Science, Vol. 15, No. 5, pp. 488～498, January, 1973.

5） Plane, D. R. and C. McMillan, Jr. （黒田　充ほか訳）：整数計画法入門, 培風館, 昭和51年6月.

6） Hu, T. C.（伊理正夫監訳）：整数計画法とネットワーク・フロー, 培風館, 昭和50年10月.

7） 森地　茂：通勤鉄道ネットワーク決定方法に関する研究, 土木学会論文報告集, 第254号, pp. 73～80, 1976年10月.

第３章　最適交通網構成問題の厳密解法に関する考察

３．１　概　　説

この章では、基本的な最適交通網構成問題を対象とし、厳密解法を考察する。アルゴリズムの改良を目的に、まず従来のアルゴリズムを検討する。そして、その結果をもとに、分枝後退法における分枝方法を提案する。さらに、アルゴリズムの改良の効果を例題を解いて調べる。

３．２では、分枝限定法と分枝後退法の２種類の厳密解法を比較する。そして、分枝後退法のScott のアルゴリズムと Hoang のアルゴリズムを検討する。つぎに、その結果からアルゴリズムの改良方法を考える。

３．３では、実行可能解までいちどに分枝する実行可能化分枝を用いるアルゴリズムを提案する。

３．４では、解集合に属する解がすべて連結網になるように分枝する連結網化分枝を用いるアルゴリズムを提案する。

３．５では、提案した実行可能化分枝と連結網化分枝の効果を調べる。これらの分枝方法を使う場合と使わない場合のアルゴリズムでいくつかの例題を解き、その結果を考察する。

３．６では、この章で得られた成果をまとめる。

３．２　厳密解法の従来のアルゴリズムの検討

３．２．１　はじめに

この節では、最適交通網構成問題の最適解を求めるために、どのようなアルゴリズムが考案されているかを述べる。そして、それらのアルゴリズムを検討し、計算時間を短縮するための改良方法を考える。

３．２．２では、分枝限定法と分枝後退法の２種類の厳密解法のアルゴリズムを比較する。そして、この研究で分枝後退法を用いる理由を示す。

３．２．３では、分枝後退法の基礎的なものである Scott のアルゴリズムを検討する。

３．２．４では、目的関数の下限値を導入した Hoang のアルゴリズムを検討する。

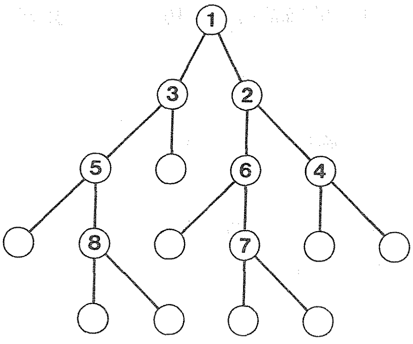
３．２．５では、従来のアルゴリズムをもとにし、計算効率を高めるにはどうすればよいかを考える。

３．２．２　分枝限定法と分枝後退法

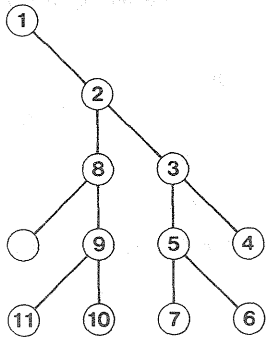
最適交通網構成問題の最適解を求めるには陰的列挙法が使われる。この陰的列挙法を用いる解法は、解の探索順序の違いにより分枝限定法（branch and bound method）と分枝後退法（branch and backtrack method）とに分類されている。[1] ここでは、まず分枝限定法と分枝後退法の特徴をあげる。そして、アルゴリズムの改良の可能性や近似解法への応用の容易さを考えたとき、どちらの解法を用いるのがよいか検討する。

陰的列挙法では系統的な解の探索を行って最適解を求める。その探索過程は探索木で表わされる。分枝限定法と分枝後退法の探索木の例を示すと、図－3.1、3.2のようになる。探索木の節点は解の集合を表わし、そこから下に出る枝はひとつの変数を選んで値を固定し、部分集合を取り出すことを表わす。この操作は分枝とよばれる。.

図－3.1､3.2で節点につけた数字は解の探索順序を表わす番号である。両者の順序の違いは、次に示す分枝規則の違いによる。



図—3.1　分枝限定法の探索木の例



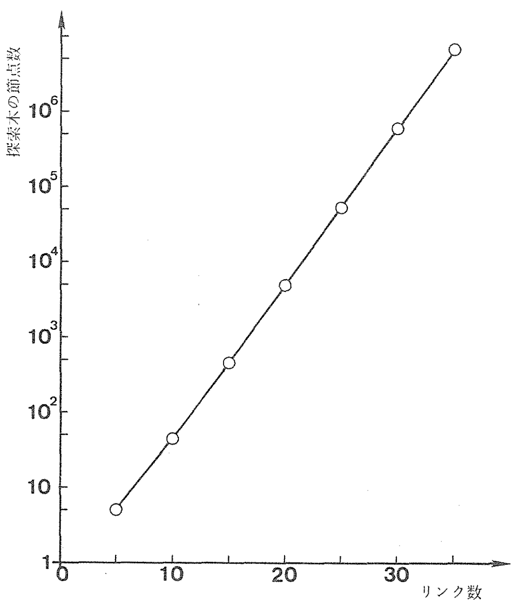
図—3.2　分枝後退法の探索木の例

分枝限定法：ある計算段階において、最小の目的関数値か目的関数の下限値を持つ解を含む節点から分枝する。

分枝後退法：最後に到達した節点から分枝する。

どちらの場合でも、すべての変数の値を固定しない状態から計算を始める。最適性の検査と実行可能性の検査を行い、可能な限り分枝を続ける。解集合に属する解がすべて実行可能である節点、最適性あるいは実行可能性の検査で解集合に最適解を含まないことが明らかな節点からは分枝しない。分枝後退法では分枝できなくなれば分枝できる節点まであともどりし、同じ操作を繰り返す。そして、すべての節点から分枝できなくなれば計算を終える。

分枝方法の違いにより、分枝限定法と分枝後退法は、それぞれ次の特徴をもつとされている。すなわち、分枝限定法は最適解を含む可能性の高い節点から調べてゆくので計算効率が高い。しかし、最適解を含む可能性のあるすべての節点の情報を保持しなくてはならないため、必要な計算機の記憶容量が大きい。これに対して、分枝後退法の計算効率は分枝限定法に劣る。けれども、どういう分枝操作を経てその節点に到達したかという情報だけがあればよく、必要な計算機の記憶容量は小さい。



図—3.3　最大ネットワークのリンク数と探索木の節点数の関係

分枝限定法を用いるか、分枝後退法を用いるか、あるいはこれらを組み合わせて使うのかは、これらの特徴を検討して決めなければならない。そこで、まず分枝限定法で必要となる計算機の記憶容量について考える。

ある計算段階において、全体の20%の数のリンクを除いた状態の節点の情報を同時に記憶する必要があるとしよう。このとき、最大ネットワークのリンク数と、探索木で情報を保持する節点数との関係を表わすと図－3.3になる。この図を見ると、ネットワークを構

成するリンクがふえれば、探索木の節点数は指数曲線状に増加することがわかる。このことから、記憶容量の増加のために分枝限定法の適用が小規模なネットワークの問題に限られることが理解できる。

分枝限定法と分枝後退法のアルゴリズムの計算効率の差を定量的に分析している例はない。しかし、分枝方法の違いから、多少の例外はあっても分枝限定法の計算効率が分枝後退法を上回ることは明らかである。

大きな記憶容量を確保したいということが、計算機システムのハードウェアに対する要求になるのに比べれば、計算時間の要求は実現しやすい。そのため、分枝限定法の必要な記憶容量が大きいという短所と、分枝後退法の計算効率が劣るという短所とを比較すると、分枝後退法の方が使いやすい場合が多いであろう。

つぎに、アルゴリズムの改良の可能性を考える。アルゴリズムを改良するには分枝方法を変えるか、最適性や実行可能性の検査方法を変えるかである。分枝方法の変更は、それぞれの基本的な分枝規則に従う範囲でなければならない。

分枝限定法は、最適解を含む可能性の高い解集合を選び、ふたつの部分集合に分割する方法である。そこで、最適解を含む可能性が高いと判断する基準を変えるか、値を固定する分枝変数を決める基準を変えれば、アルゴリズムを変更できる。これに対し、分枝後退法で分割する解集合は最後に到達した節点が表わすものである。そのため、アルゴリズムを変更するには分枝変数の選択基準を変えるだけである。

けれども、分枝操作の自由度は分枝後退法の方が高い。分枝変数として選んだリンクをネットワークに含む解の集合と、ネットワークに含まない解の集合とのふたつの部分集合に解集合を分割することが分枝である。このとき、分枝限定法では分枝変数は必ずひとつでなければならない。複数の分枝変数を選んで同時にその値を固定することは、1変数ずつの分枝を繰り返すのと同じである。これは、最適解を含む可能性の高い解集合を分割するという分枝規則に反するのである。しかし、分枝後退法では最後に到達した節点の表わす解集合を分割するのであるから、このような分枝操作も可能である。すなわち、分枝限定法では解集合をつねに同じ大きさのふたつの部分集合に分けなければならないが、分枝後退法では任意の大きさの部分集合を切り出す分枝操作ができるのである。

最適性の検査は、解集合に最適解が含まれる可能性があるかどうかを調べるものである。そして、実行可能性の検査は、解集合に実行可能解が含まれるかどうかを調べるものである。これらは、解集合がどのような分枝を経て作られたか、ということには関係しない。そのため、分枝限定法と分枝後退法のアルゴリズムの改良の可能性を比較するとき、これらを考慮しなくてもよい。

このように、アルゴリズムの改良の可能性を比較するには、分枝方法をどのように変えられるかだけを考えればよい。分枝限定法と分枝後退法では変更できる部分が異なるが、改良の可能性には大差ないであろう。

つぎに、近似解法への応用の容易さを調べる。厳密解法を近似解法に直す場合、解の探索範囲を限定して計算量を減らす方法がある。厳密解法では最適解が含まれないことが明らかか、実行可能解が含まれないことが明らかでない限り、解集合を探索範囲から除くことはない。しかし、

近似解法では最適解が含まれる可能性が低いと判断すれば、解集合を探索範囲から除いてよい。分枝限定法も分枝後退法も、このようにアルゴリズムを変更することは容易である。

近似解法への応用のもうひとつの方法は、厳密解法を既存の近似解法と組み合わせて使い、解の精度を向上させるものである。分枝限定法と分枝後退法は、このような方法にも同じように使える。

厳密解法を近似解法に直したり、他の近似解法と組み合わせることにおいて、分枝限定法と分枝後退法に差はない。しかし、作成した近似解法を使って問題を解くときに、どちらを用いているかで違いが生じる。近似解法は規模の大きいネットワークの問題に用いることが前提となっている。大規模問題に適用すれば、分枝限定法の部分では必要な記憶容量が大きい。厳密解法として用いる場合よりは小さいであろうが、大規模な問題に適用しにくいことは近似解法の大きな欠点である。したがって、近似解法への応用に関しては分枝後退法の方が優れている。

これまでに述べてきたことをまとめれば、次のようになる。

① 分枝限定法の必要な記憶容量が大きいという短所と、分枝後退法の計算効率が劣るという短所を比べれば、分枝後退法の方が使いやすい。

② 分枝限定法と分枝後退法ではアルゴリズムの変更できる部分は異なるが、改良の可能性には大きな差はない。

③ 分枝限定法を近似解法に応用した場合、必要な記憶容量が大きいために大規模な問題に適用できないことが予想できる。

これらのことから、本研究では厳密解法に分枝後退法を採用し、計算時間を短縮する改良を行う。

3.2.3 Scottのアルゴリズム

分枝後退法のアルゴリズムとしてよく知られているものに、Scott の backtrack programming algorithm[2)]がある。これは、分枝後退法のもっとも単純なもののひとつであろう。そこで、分枝後退法の基本的なアルゴリズムとして Scott のアルゴリズムを検討する。

問題の表現は異なるが、Scott のアルゴリズムは最適交通網構成問題の基本問題を解くためのものである。すなわち、ネットワークの総延長の制約下でノード間の最短距離の合計を最小にするリンクの組み合わせを求めるのである。基本問題は第2章で問題2.2として定式化したが、ノード数とリンク数を明確にして定式化し直せば、次のようになる。ただし、リンクに向きはないものとし、どちらの方向へも通れるとする。

問題3.1

$$\min \quad Z=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}(\boldsymbol{x}) \tag{3.1}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} l_k x_k \leqq L^c \tag{3.2}$$

$$x_k = 0 \ or \ 1 \quad (k=1, 2, \ldots, m) \tag{3.3}$$

ここに、Z：ノード対の最短距離の合計

d_{ij}：ノード ij 間の最短距離で、経路がないときは無限大とする。

$\boldsymbol{x}=\{x_1, x_2, \ldots, x_k, \ldots, x_m\}$

：ネットワークの状態を表わし、リンクkがネットワークに含まれないとき $x_k=0$ 、

ネットワークに含まれるとき $x_k=1$ とする。

l_k：リンク k の長さ

L^c：ネットワークの総延長の制約値

n：ノード数

m：リンク数

Scott のアルゴリズムはリンクの順序に基づいた探索木をあらかじめ作っておき、それに従って最適解を探すものである。問題３.１の表現を用いてアルゴリズムの概要を示すと、次のようになる。

ステップ１：リンク k の両端のノードを i_k 、j_k $(i_k<j_k)$とする。 $h<k$ ならば、$i_h<i_k$ または $i_h=i_k$ で $j_h<j_k$ となるようにリンク番号をつける。

ステップ２：最適解の目的関数の上限値Z^*に任意の大きな値を与える。

ステップ３：すべての変数を自由変数、すなわち $x_k \leftarrow -1$ とする。

ステップ４： $x_k \neq 0$ であるリンクからなるネットワークの目的関数値 Z を求める。

ステップ５： $Z<Z^*$ ならばステップ６へ、 $Z \geqq Z^*$ ならばステップ９へ進む。

ステップ６：Zを求めたネットワークが実行可能ならばステップ７へ、実行可能でなければステップ８へ進む。

ステップ７：Z^* を Z で、$\boldsymbol{x}^*$ を $\boldsymbol{x}=\{|x_k|\}$ で置き換え、ステップ９へ進む。

ステップ８： $k=min\{h|x_h=-1\}$ により分枝変数を定め、 $x_k \leftarrow 0$ としてステップ４へもどる。もし自由変数が残っていなければ、ステップ９へ進む。

ステップ９：その節点への分枝変数 x_k の値が０ならばステップ10へ、１ならばステップ11へ進む。

ステップ10：ひとつ前の節点へあともどりして $x_k \leftarrow 1$ とし、もう一方の節点へ分枝する。ステップ４へもどる。

ステップ11：$x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。

ステップ12：その節点が探索を開始した節点ならば計算を終える。そうでなければステップ９へもどる。

アルゴリズムの中では自由変数であることを$x_k=-1$で表わしている。計算を終えたときの $\boldsymbol{x}^*$ が最適解である。

分枝変数はステップ８で決めるが、これを見れば探索木がリンク番号によって決まることがわかる。リンク番号はステップ１で決めているので、Scott のアルゴリズムでの探索木はノード番号のつけ方で定まることになる。すなわち、ノード番号のつけ方で計算効率が左右されるのである。

このアルゴリズムでは、ステップ５で最適性の検査が行われている。しかし、実行可能性は検査されていない。ステップ６は、その節点の解集合に属するすべての解が実行可能であるかどうかを調べるためのものであり、解集合に実行可能解が含まれるかどうかをみるものではない。このため、解集合に実行可能解が含まれていなくても分枝を繰り返すことになる。

このように、Scott のアルゴリズムは分枝後退法の概念を理解するためには役立つが、計算効率は悪い。

3.2.4 Hoang のアルゴリズム

Hoang の branch search algorithm[3]は、探索木上の節点の解集合に含まれる実行可能解の目的関数の下限値を求める手続を導入したことに特徴がある。下限値を用いることにより、最適性の検査を強力にしている。そして、その一方で最短経路探索を含む目的関数の計算回数を減らしている。さらに、下限値を求めるための補助問題の解の性質を利用して分枝変数を決めているので、探索木の作成方法は合理的である。

目的関数の下限値の計算は、各リンクを除くことによる目的関数の増加の下限値をもとにしている。解集合の中で目的関数の増加の下限値の合計が最小である実行可能解を求め、解集合に対する目的関数の下限値を定めようとするのである。このような実行可能解を求める問題はナップザック問題になり、解くのは容易でない。そこで、下限値を簡単に計算するために整数制約をはずした線形計画問題を解いている。

この線形計画問題の解は1変数だけが小数成分を持ち、他の変数は0か1のどちらかになる。そこで、小数成分を持つ変数に対応するリンクを選び、分枝変数にしている。このように分枝変数を選ぶと、分枝を行えば必ず目的関数の下限値が大きくなるからである。

問題3.1の表現を用いて Hoang のアルゴリズムの概要を示すと、次のようになる。

ステップ1：近似解法によって初期解 $\boldsymbol{x}^*=\{x_1^*, x_2^*, \ldots, x_m^*\}$ を求め、その目的関数値を Z^*、ネットワーク長を L^* とする。

ステップ2：各リンクを除くことによる目的関数の増加の下限値を求める。

ステップ3：すべての変数を自由変数、すなわち $x_k \leftarrow -1$ とする。

ステップ4：$x_k=0$ あるいは $x_k=1$ の固定変数を制約に加え、下限値を求めるための線形計画問題を解き、目的関数の下限値 Z^l を計算する。

ステップ5：$Z^l \leqq Z^*$ ならばステップ6へ、$Z^l > Z^*$ ならばステップ10へ進む。

ステップ6：線形計画問題の解で小数成分を持つ変数を選び、分枝変数 x_r とする。$x_r \leftarrow 0$ として分枝する。

ステップ7：$x_k \fallingdotseq 0$ であるリンクからなるネットワークが実行可能であれば、ステップ8へ進む。そうでなければステップ4へもどる。

ステップ8：$x_k \fallingdotseq 0$ であるリンクからなるネットワークの目的関数値 Z とネットワーク長 L を求める。

ステップ9：(Z, L) が辞書編集的に (Z^*, L^*) より小さければ Z^* を Z で、L^* を L で、$\boldsymbol{x}^*$ を $\boldsymbol{x}=\{|x_k|\}$ で置き換える。ステップ10へ進む。

ステップ10：その節点への分枝変数 x_r の値が0ならばステップ11へ、1ならばステップ12へ進む。

ステップ11：ひとつ前の節点へあともどりして $x_r \leftarrow 1$ とし、もう一方の節点へ分枝する。ステップ4へもどる。

ステップ12：$x_r \leftarrow -1$ とし、ひとつ前の節点へあともどりする。

ステップ13：その節点が探索を開始した節点ならば計算を終える。そうでなければステップ10へもどる。

計算を終えたときの $\boldsymbol{x}^*$ が最適解である。ステップ1で近似解法を用いて初期解を求めるのは、目的関数の上限値を与えて分枝範囲をせばめるためである。ステップ9での暫定最適解との比較で明らかなように、Hoang のアルゴリズムでは目的関数値が同じ解がある場合、ネットワーク長の小さい解を採択している。

このアルゴリズムでは、解集合に実行可能でない解を含む探索木の節点に対しては目的関数値を計算しない。そのかわりに目的関数の下限値を求めるのである。下限値の計算には線形計画問題を解かなければならないが、シンプレックス法を使わずに、簡単に解を得ることができる。そのため、各節点における計算量は Scott の場合よりも少ない。

Scott のアルゴリズムでの最適性の検査は、節点の表わす解集合の中でもっとも目的関数値の小さいものと、目的関数の上限値を比較するものであった。これに対して Hoang のアルゴリズムでは、解集合に含まれる実行可能解の目的関数の下限値を目的関数の上限値と比較している。問題の制約条件を考慮している点において、Hoang のアルゴリズムが優れている。

このように、Hoang のアルゴリズムは計算効率を高める改良がいくつかなされている。目的関数の下限値を求めるとき、リンクをネットワークから除けば目的関数値はどれだけ増加するかを考え、最適交通網構成問題の特性をうまく利用している。アルゴリズムを改良しようとするとき、汎用的な解法を対象とする問題に合わせて直すことが必要である。

ここでのアルゴリズムの記述は、Hoang の示したものを問題３．１の表現にあわせて書き改めたものである。計算内容は変わっていない。詳しく調べると、このアルゴリズムには不十分な箇所がある。たとえば下限値を求めるために線形計画問題を解いたとき、小数成分を持つ変数がない場合が起こり得る。しかし、このときの分枝変数の選択方法は述べられていない。さらに、線形計画問題の実行可能解がない場合もある。このときは、解集合に属する解はすべて実行不可能なので、それ以上の分枝は必要なく、あともどりすればよい。けれども、これもアルゴリズムでは触れられていない。これは、実行可能性の検査という概念を明確に持っていなかったからではないかと考えられる。

Hoang のアルゴリズムは記述に不十分な点はあるが、Scott のアルゴリズムに比べれば計算効率が改善されている。

３．２．５　アルゴリズムの改良方法

分枝後退法の基礎的なものである Scott のアルゴリズムに対し、Hoang のアルゴリズムは計算時間を短くするための改良がなされたものである。ここでは、この改良されたアルゴリズムをもとにし、計算時間をさらに短縮するにはどうすればよいのか検討する。

分枝後退法の計算効率を高めるには、次のことを考えればよい。

①　探索木での分枝を早い段階で打ち切り、むだな部分を探索しない。

②　分枝操作に必要な計算量を減らす。

探索範囲を最適解を含む可能性のある小さな部分に限定し、各段階での計算を簡単にすればよいのである。これらは必ずしも同時に実現できるとは限らず、分枝操作を簡単にすれば探索範囲が広がってしまうこともある。したがって、それぞれのバランスを考え、総合すれば計算時間が短くなるようにしなければならない。分枝後退法のアルゴリズムの主な構成要素は最適性の検査

と実行可能性の検査、それに分枝変数の選択である。ここにあげたことを考えたうえで、アルゴリズムの各部分の改良を図る。

Hoang のアルゴリズムでは、目的関数の下限値を用いて最適性の検査を行っている。これと、近似解で目的関数の上限値を与えていることが分枝範囲の限定に役立っている。3.2.4で述べたようにアルゴリズムでは明確でないが、目的関数の下限値を求めるための線形計画問題を解くことで、実行可能性を調べることもできる。そして、分枝を行えば目的関数の下限値が大きくなるように分枝変数を選んでいる。これも分枝範囲を限定することを考えたものといえよう。分枝操作の計算量を減らすことは、分枝のために目的関数値の計算を行わない点で考慮されているといえる。

Hoang の行った改良を進めるのならば、目的関数の上限値と下限値を強くすることが考えられる。目的関数の上限値の初期値を強めるには、最適解に近い解が得られる近似解法を使用すればよい。しかし、より良い近似解を得るには計算時間を費やすことが必要である。近似解を求めるための計算時間の増加量と、強力な上限値を用いることによる計算時間の減少量とを比べなければならない。

Scott は簡単な2種類の近似解法の計算例で、最適解に対する近似解の目的関数値の損失率が10.4%以下、あるいは2.3%以下であったと報告している。[2] それに対し、上限値と比較する下限値の方は、このような水準まで強力にすることは困難であろう。そこで、初期解を求める近似解に計算時間をかけても、その効果は期待できない。すなわち、目的関数の上限値を強化することは大きな改良にならないであろう。

目的関数の下限値を強くするには、異なる観点から下限値を定義しなければならない。その場合、部分ネットワークに分割するとか、除くリンクの組み合わせを用いて下限値を求める方法が考えられる。しかし、いずれも計算は Hoang の用いている下限値より複雑になる。そして、リンクの組み合わせを使えば必要な記憶容量が大きくなる。このように、目的関数の下限値を強くすることはできるが、分枝後退法に適する簡潔なものはできないであろう。なお Hoang は計算途中のネットワークをあたかも最大ネットワークのように考えることにより、下限値を強めることを提唱しているが、これは下限値の定義にかかわる本質的なことではない。

以上に述べたように、Hoang の行った改良を強化する方法には見込みがない。そこで、Hoang のアルゴリズムでは考慮されていない改良方法を考える。

分枝後退法では任意の大きさの部分集合を切り出す分枝操作ができることを3.2.2で述べた。この分枝方法は、1回の分枝操作で複数の分枝変数の値を固定し、探索木上を複数段階進むものである。Hoang のアルゴリズムの分枝方法は1変数ずつ固定していくものである。そこで、分枝方法を変えることにより、分枝操作に必要な計算量が減り、計算時間を短くできるかもしれない。

Scott のアルゴリズムは通常の0－1整数計画問題の解法そのままといえる。それに対し、Hoang のアルゴリズムは目的関数の下限値の導入において、最適交通網構成問題の特性を利用している。このように、対象とする問題に合わせたアルゴリズムにすることが必要である。そして、Hoang のアルゴリズムでも考慮されていない最適交通網構成問題の特性がある。それは、

最適解ではネットワークが必ず連結でなければならないことである。これはすでに Scott が指摘しているが、アルゴリズムには組み込まれていない。広い意味では最適性の検査に含まれるが、連結性の検査を設ければよい。さらに、ネットワークが非連結な解を作らないような分枝方法を用いれば、解の探索範囲が狭くなり、計算効率が高くなるであろう。

以上の検討に基づき、次の 2 点でアルゴリズムの改良を図ることにする。

① 一度に複数の分枝変数の値を固定する分枝操作の採用

② ネットワークが非連結になる解をつくらない分枝方法の採用

3.2.6 むすび

この節では最適交通網構成問題の従来のアルゴリズムを検討し、その改良方法を考察した。最初に分枝限定法と分枝後退法の 2 種類の分枝方法を比較した。つぎに、Scott と Hoang の分枝後退法のアルゴリズムを詳しく調べ、さらに計算時間を短縮する方法を考えた。

最適交通網構成問題の厳密解法は、分枝限定法と分枝後退法とに大別される。これらをいくつかの点で比べると、分枝限定法には必要な記憶容量が大きいという欠点があった。近似解法へ応用した場合にもこの欠点は問題になる。このことから、この研究においては基本的な分枝方法に分枝後退法を用いることを決めた。

従来のアルゴリズムは、基礎的な Scott のものと計算効率を改善した Hoang のものを検討した。Scott のアルゴリズムは通常の 0 − 1 整数計画問題の解法と大差ないが、Hoang のアルゴリズムは最適交通網構成問題の特性をうまく利用していた。しかし このアルゴリズムをさらに改良しようとする場合、Hoang の行った改良を強化する方法には見込みがなさそうであった。そこで、一度に複数の分枝変数の値を固定する分枝操作の採用と、ネットワークが非連結になる解を作らない分枝方法の採用を考えた。

3.3 実行可能化分枝を用いるアルゴリズムの提案[4)]

ここでは、複数の分枝変数の値を一度に固定する分枝操作を用いるアルゴリズムを提案する。

分枝操作の繰り返しを少なくするには、分枝を打ち切る節点まで 1 回の分枝操作で分枝するのがよい。分枝後退法で分枝を打ち切るのは、探索木の表わす解集合が次のどれかになったときである。

① 解集合に属する解がすべて実行可能になったとき

② 解集合には最適解が含まれないことがわかったとき

③ 解集合には実行可能解が含まれないことがわかったとき

したがって、最適性の検査と実行可能性の検査に合格したときに実行可能解だけからなる解集合の節点まで分枝すれば、必ずその節点で分枝を打ち切れる。よって、そのような解集合すなわち実行可能解集合まで一度に分枝する操作を用いることにする。この分枝操作を行えば解集合に属する解がすべて実行可能になるので、これを実行可能化分枝と呼ぶことにする。

解集合に属する解がすべて実行可能解であるとき、その中で必要なのは目的関数値が最小のものだけである。問題 3．1 では、リンクを除けば目的関数値は同じか大きくなる。そのため、目的関数値が最小の解は自由変数のままの変数の値を 1 にすることで得られ、この解が解集合を代

表する。これは Scott や Hoang のアルゴリズムと同じである。そして、このことは他の実行可能解集合の部分集合である解集合まで分枝しないようにしなければならないことを意味している。

実行可能解まで分枝するとき、その実行可能解は目的関数値が小さいことが望ましい。そうすれば、探索の早い段階で最適解が見つかるか、あるいは目的関数の上限値が小さくなって分枝範囲が狭くなるからである。よって、分枝の基本的な考え方は、分枝前の節点の表わす解集合の中で、目的関数値が最小である可能性の高い実行可能解が代表する部分集合まで一度に分枝するということになる。そこで、そのような実行可能解を見つけるために、目的関数の下限値を求めるための補助問題を再検討する。この補助問題は次に示すものである。

補助問題３．１．１

$$\min \quad F=\sum_{k=1}^{m} f_k y_k \tag{3.4}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} l_k y_k \geqq L^{max}-L^c \tag{3.5}$$

$$y_k=0 \ or \ 1 \quad (k=1,2,\ldots,m) \tag{3.6}$$

ここに、 F : 実行可能にすることによる目的関数 Z の増加の下限値

f_k : リンク k をネットワークから除くことにより生じる目的関数 Z の増加の下限値

$L^{max}=\sum_{k=1}^{m} l_k$

: 最大ネットワークの総延長

式（３．４）の補助問題の目的関数は、リンクを除くことによって生じる目的関数の増加の下限値を最小にすることを表わしている。式（３．５）はネットワーク長の制約を満たすまで最大ネットワークからリンクを除くことを表わす。ただし、最大ネットワークとは検討対象のすべてのリンクを含むネットワークである。

補助問題３．１．１は０－１整数計画問題で、ナップザック問題と呼ばれる種類のものである。解を求めるにはかなりの計算が必要なので、Hoang は式（３．６）の整数制約を実数制約にゆるめた線形計画問題をかわりに解いている。これが補助問題３．１．２である。

補助問題３．１．２

$$\min \quad F'=\sum_{k=1}^{m} f_k y_k \tag{3.7}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} l_k y_k \geqq L^{max}-L^c \tag{3.8}$$

$$0 \leqq y_k \leqq 1 \quad (k=1,2,\ldots,m) \tag{3.9}$$

問題３．１の目的関数の下限値 Z^l を求めるには、探索木のその節点までで値を固定されている変数に対応して $y_k=0$ 、あるいは $y_k=1$ の制約式を加えて補助問題３．１．２を解く。そうすれば、最大ネットワークの目的関数値を Z^0 として、式（３.10）で下限値が計算できる。

$$Z^l=Z^0+F' \tag{3.10}$$

補助問題３．１．２の最適解は固定変数の制約式がない場合、リンクを式（３.11）を満たす

ように並べかえれば、式（3.12）のようになる。ただし、r は式（3.13）を満たす最小の整数である。

$$\frac{f_1}{l_1} \leqq \frac{f_2}{l_2} \leqq \cdots\cdots \leqq \frac{f_m}{l_m} \tag{3.11}$$

$$\begin{cases} y_k = 1 & if \quad k < r \\ y_k = 0 & if \quad k > r \\ y_r = \dfrac{L^{max} - L^c - \sum_{k=1}^{r-1} l_k}{l_r} & \end{cases} \tag{3.12}$$

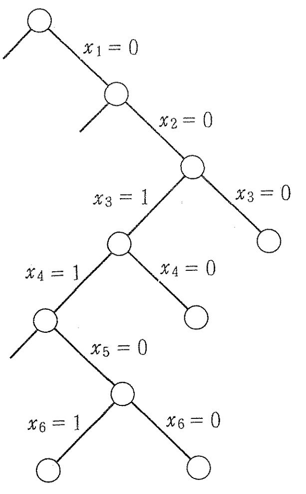
$$\sum_{k=1}^{r} l_k \geqq L^{max} - L^c \tag{3.13}$$

ここで、式（3.12）のかわりに次の式（3.14）の解を考える。

$$\begin{cases} y_k = 1 & if \quad k \leqq r \\ y_k = 0 & if \quad k > r \end{cases} \tag{3.14}$$

この解は明らかに補助問題3.1.1の実行可能解である。そして、式（3.13）の両辺の差が小さいときには最適解であることが多いと考えられる。補助問題3.1.1の最適解は、補助問題を解いた節点の表わす解集合の中で、もっとも目的関数の下限値の小さい実行可能解である。そこで、式（3.15）を使って分枝操作を行えば、解集合の中で目的関数値が最小である可能性の高い実行可能解へ達することができる。なお、式（3.15）は x_k に0を代入することを表わしている。

$$x_k \leftarrow 0 \quad for \quad k \leqq r \tag{3.15}$$



図―3.4　実行可能化分枝を用いるアルゴリズムの探索木の例

分枝変数として値を割り当ててゆく順序を式（3.11）を満たすリンク番号順とすれば、探索は図－3.4のようになる。すなわち、ネットワーク長の制約式を満たすまで自由変数からリンク番号順に分枝変数を選び、0を割り当てることになる。リンク番号のつけ方が異なるが、探索木の構造はScott のアルゴリズムの場合と同じである。

実行可能化分枝で一度に何段階かの分枝を行っても、あともどりするのは1段階ずつである。実行可能化分枝を用いるアルゴリズムは次のようになる。

ステップ1：各リンクを除くことによる目的関数の増加の下限値 f_k を求める。

ステップ2： $f_1/l_1 \leqq f_2/l_2 \leqq \ldots \leqq f_m/l_m$ となるようにリンクを並べかえる。

ステップ3：近似解法によって初期解 $\boldsymbol{x}^* = \{x_1^*, x_2^*, \cdots, x_m^*\}$ を求め、その目的関数値を Z^*、ネットワーク長を L^* とする。

ステップ4：すべての変数を自由変数、すなわち $x_k \leftarrow -1$ とする。

ステップ５：$x_k=0$ あるいは $x_k=1$ の固定変数を制約に加え、補助問題３．１．２を解く。最適解が求まればステップ６へ、実行可能解がなければステップ11へ進む。

ステップ６：目的関数の下限値 Z^l を計算する。

ステップ７：$Z^l \leqq Z^*$ ならばステップ８へ、$Z^l > Z^*$ ならばステップ11へ進む。

ステップ８：自由変数からリンク番号の小さい順に分枝変数 x_k を選び、ネットワーク長が制約値以下になるまで $x_k \leftarrow 0$ として分枝する。

ステップ９：$x_k \neq 0$ であるリンクからなるネットワークの目的関数値 Z とネットワーク長 L を求める。

ステップ10：(Z, L) が辞書編集的に (Z^*, L^*) より小さければ、Z^*を Z で、L^*を L で、$\boldsymbol{x}^*$ を $\boldsymbol{x}=\{|x_k|\}$ で置き換える。ステップ11へ進む。

ステップ11：その節点への分枝変数 x_k の値が０ならばステップ12へ、１ならばステップ13へ進む。

ステップ12：ひとつ前の節点へあともどりして $x_k \leftarrow 1$ とし、もう一方の節点へ分枝する。ステップ５へもどる。

ステップ13：$x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。

ステップ14：その節点が探索を開始した節点ならば計算を終える。そうでなければステップ11へもどる。

計算を終えたときの $\boldsymbol{x}^*$ が最適解である。

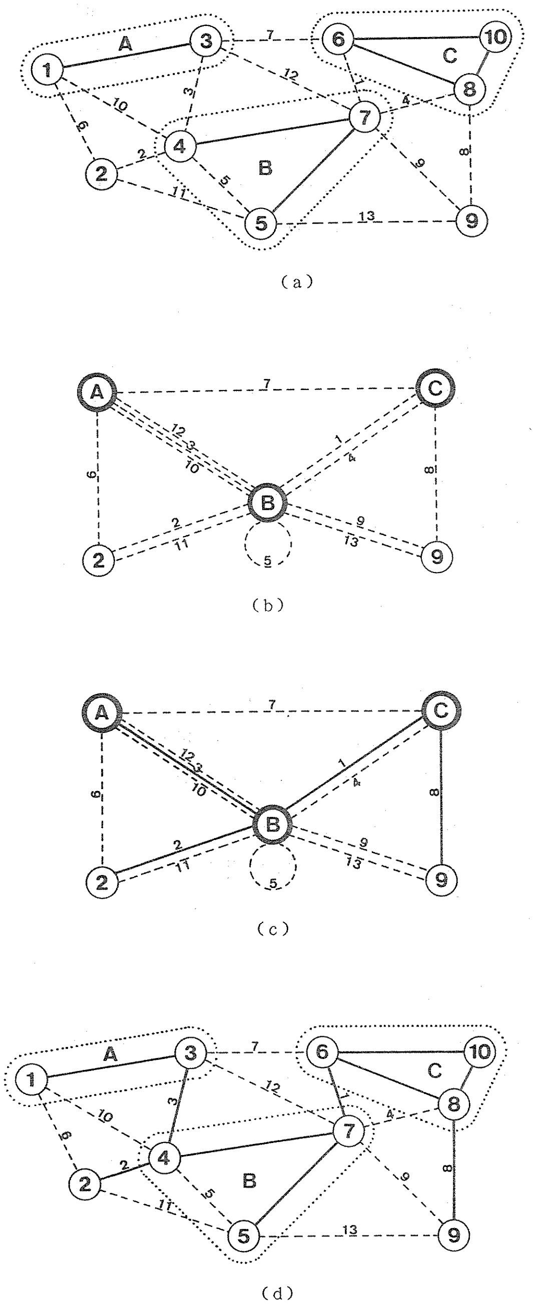
３．４　連結網化分枝を用いるアルゴリズムの提案[4)]

ここでは、探索木の分枝範囲を解のネットワークが連結である部分に限定する方法を示す。そして、解のネットワークが連結になるように分枝した後、最適解の探索を行うアルゴリズムを提案する。

ノード $i\,j$ 間に経路がない場合、最短距離 d_{ij} は無限大であるとした。したがって、そのネットワークの目的関数値も無限大である。よって、最適解のネットワークではすべてのノード間に経路がなければならない。連結網でなければならないのであり、非連結網が最適解になることはない。しかし、Scott や Hoang のアルゴリズムには、解集合に連結網である実行可能解が含まれているかどうかを調べる過程はない。そのため、とくに Hoang のアルゴリズムでは目的関数の下限値だけで最適性を判断して分枝を行うので、非連結網である解の探索を繰り返すおそれがある。

ネットワーク長の制約値が小さいときは実行可能解のネットワークに含まれるリンクが少なく、非連結網である割合が大きい。このような場合、解集合に属する実行可能解がすべて非連結網であることがわかれば分枝を打ち切ってよい。そうすればむだな分枝を繰り返さないため、計算効率を向上させることができる。これを行うのが連結性の検査である。

解集合に連結網である実行可能解が含まれるかどうかを調べるには、解集合に属する解のうち連結網でネットワーク長が最短のものが制約式を満たすかどうかをみればよい。そのようなネットワーク、すなわち最短連結網を探す方法を簡単な例を使って説明する。



図—3.5　最短連結網の例

探索木の節点でのリンクの状態は、ネットワークに含めると固定されているか、ネットワークに含めないと固定されているか、あるいはまだどちらとも決められていないかのどれかである。変数で表わすと$x_k=1, 0, -1$である。いま、ある計算段階のネットワークが**図－3.5**（a）であるとする。図の実線は $x_k=1$ 、破線は $x_k=-1$ のリンクで、$x_k=0$ のリンクは省いてある。

ノード1、3、ノード4、5、7、ノード6、8、10はそれぞれ $x_k=1$ のリンクで結ばれている。これらのノードは探索木上の現在の節点からあともどりしないかぎり、たがいに結ばれたままである。つまり、解集合に属する解ではつねに連結である。そこで、これらの連結成分をひとつのノードとみなし、ノードA、B、Cとする。そうすれば、**図－3.5**（a）は**図－3.5**（b）のように書き換えることができる。

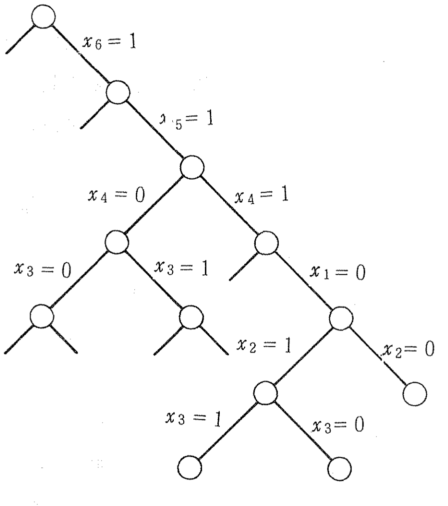
ネットワークを構成するすべてのノードをつなぎ、長さのもっとも短いものは最短生成木（minimal spanning tree）である。よって、**図－3.5**（b）の最短生成木をさがせば、それが解集合の最短連結網を与える。**図－3.5**ではリンクにつけた数字がリンク長の短い順を表わしている。そこで、最短生成木は**図－3.5**（c）に示すものになる。これを元のネットワークにもどせば**図－3.5**（d）になる。

こうして求めた最短連結網のネットワーク長が制約値よりも大きいならば、解集合に連結網である実行可能解は含まれていない。それ以上の分枝は必要なく、あともどりすればよい。この操作が連結性の検査である。

連結性の検査をさらに進め、積極的に連結網である実行可能解へ分枝すれば、計算効率はより向上する。ある節点で求めた最短連結網のネットワーク長が制約値以下のとき、最短連結網を含む解からなる解集合に相当する節点へ分枝すればよい。そうすれば、あともどりしないかぎり解集合には非連結網である解は含まれない。この分枝操作を行えば解集合に属する解がすべて連結網になるので、これを連結網化分枝と呼ぶことにする。

連結網化分枝の方法をまとめれば、次のようになる。まず $x_k=1$ のリンクで結ばれた連結成分をひとつのノードとみなしたネットワークで、$x_k=-1$ のリンクからなる最短生成木を探す。こうして求めた最短連結網のネットワーク長が制約値以下ならば、いまの最短生成木を構成するリンクを分枝変数として選び、値を1に固定する。この連結網化分枝も3.3で述べた実行可能化分枝と同じように、一度に複数の分枝変数の値を固定することになる。分枝変数に選ぶ順序はリンクの長さの短い順とする。それは、分枝を繰り返しても長さの短い生成木がネットワークに含まれているようにするためである。

連結網化分枝を行って解集合に属する解がすべて連結網になれば、これまでと同じ分枝方法で最適解を探せばよい。Hoang のアルゴリズムと同じ分枝方法でも、実行可能化分枝を使ってもよい。ただし、連結網化分枝とこれらの分枝方法では分枝変数の値の固定順序が異なる。そのため、探索木は図－3.6に例を示すように2段階の構造になる。



図―3.6　連結網化分枝を用いるアルゴリズムの探索木の例

連結網化分枝と実行可能化分枝を用いるアルゴリズムは次のようになる。

ステップ1：各リンクを除くことによる目的関数の増加の下限値 f_k を求める。

ステップ2： $f_1/l_1 \leqq f_2/l_2 \leqq \cdots \leqq f_m/l_m$ となるようにリンクを並べかえる。

ステップ3：近似解法によって初期解 $\boldsymbol{x}^*=\{x_1^*, x_2^*, \cdots, x_m^*\}$ を求め、その目的関数値を Z^*、ネットワーク長を L^* とする。

ステップ4：すべての変数を自由変数、すなわち $x_k \leftarrow -1$ とする。

ステップ5：節点の表わす解集合の中での最短連結網を求め、そのネットワーク長を L^S とする。最短連結網が求まればステップ6へ、連結網がなければステップ8へ進む。

ステップ6： $L^S \leqq L^C$ ならばステップ7へ、$L^S > L^C$ ならばステップ8へ進む。

ステップ7：最短連結網を構成するリンクで $x_k=-1$ のものをリンク長の短い順に分枝変数として選び、 $x_k \leftarrow 1$ として分枝する。ステップ12へ進む。

ステップ8：その節点への分枝変数 x_k の値が1ならばステップ9へ、0ならばステップ10へ進む。

ステップ9：ひとつ前の節点へあともどりして $x_k \leftarrow 0$ とし、もう一方の節点へ分枝する。ステップ5へもどる。

ステップ10： $x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。

ステップ11：その節点が探索を開始した節点ならば計算を終える。そうでなければステップ8へもどる。

ステップ12： $x_k=0$ あるいは $x_k=1$ の固定変数を制約に加え、補助問題3.1.2を解く。

最適解が求まればステップ13へ、実行可能解がなければステップ18へ進む。

ステップ13：目的関数の下限値 Z^l を計算する。

ステップ14：$Z^l \leqq Z^*$ ならばステップ15へ、$Z^l > Z^*$ ならばステップ18へ進む。

ステップ15：自由変数からリンク番号の小さい順に分枝変数 x_k を選び、ネットワーク長が制約値以下になるまで $x_k \leftarrow 0$ として分枝する。

ステップ16：$x_k \neq 0$ であるリンクからなるネットワークの目的関数値 Z とネットワーク長 L を求める。

ステップ17：(Z, L) が辞書編集的に (Z^*, L^*) より小さければ、Z^*を Z で、L^* を L で、$\boldsymbol{x}^*$ を $\boldsymbol{x}=\{|x_k|\}$ で置き換える。ステップ18へ進む。

ステップ18：その節点への分枝が連結網化分枝によるものならばステップ8へもどる。そうでなければステップ19へ進む。

ステップ19：その節点への分枝変数 x_k の値が0ならばステップ20へ、1ならばステップ21へ進む。

ステップ20：ひとつ前の節点へあともどりして $x_k \leftarrow 1$ とし、もう一方の節点へ分枝する。ステップ12へもどる。

ステップ21：$x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。ステップ18へもどる。

計算を終えたときの $\boldsymbol{x}^*$ が最適解である。

3.5　計算例によるアルゴリズムの比較

3.5.1　はじめに

この節では、3.3で提案した実行可能化分枝と3.4で提案した連結網化分枝の採用の効果を調べる。それぞれの分枝方法を用いるかどうかで4種類のアルゴリズムを作成し、例題を解く。そして、主に計算時間を比較する。

3.5.2では、比較の対象とするアルゴリズムを説明する。

3.5.3では、計算に用いる例題を示す。

3.5.4では、例題の計算結果を示す。そして、実行可能化分枝の効果と連結網化分枝の効果を考察する。

3.5.2　比較の対象とするアルゴリズム

例題を解いてアルゴリズムを比較する目的は、提案した分枝方法の採用による計算時間短縮の効果を調べることである。そこで、実行可能化分枝を用いる場合と用いない場合、および連結網化分枝を用いる場合と用いない場合のアルゴリズムを比較の対象とする。そうすると、分枝方法の組み合わせにより**表－3.1**に示す4種類のアルゴリズムができる。ただし、実行可能化分枝を用

表―3.1　比較の対象とするアルゴリズム

連結網化分枝	実行可能化分枝	
	用いない	用いる
用いない	アルゴリズムE1	アルゴリズムE2
用いる	アルゴリズムE3	アルゴリズムE4

いないときは、Hoang のアルゴリズムと同じ分枝方法を使うものとする。

アルゴリズムE1は、実行可能化分枝も連結網化分枝も使わない。すなわち、Hoang のアルゴリズムそのままの分枝方法を用いるものである。Hoang の記述で明確でなかった部分は次のようにする。補助問題3．1．2を解いたときに小数成分を持つ変数がない場合も、式（3．13）を使って分枝変数を決める。そして、補助問題の実行可能解がないときは、3．3の実行可能化分枝を用いるアルゴリズムと同じようにあともどりする。

アルゴリズムE2は、実行可能化分枝を使うが、連結網化分枝は使わない。つまり、3．3で示したアルゴリズムである。

アルゴリズムE3は、実行可能化分枝は使わないが、連結網化分枝を使用する。解集合がすべて連結網になってからは、アルゴリズムE1と同じ分枝方法になる。

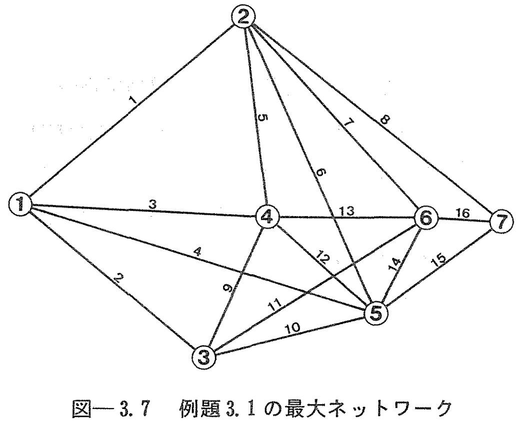
アルゴリズムE4は、実行可能化分枝と連結網化分枝の両方を用いる。すなわち、3．4で示したアルゴリズムである。

初期解を求める近似解法には Hoang が使ったものを用いる。これは、Scott の backward solution algorithm[2]からネットワーク変換操作を除いたものである。

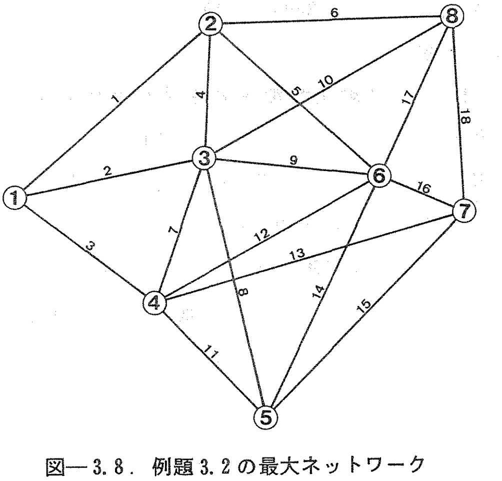
Hoang は目的関数の下限値を強める方法を提案している。リンクをいくらか除いた状態を最大ネットワークのようにみなして目的関数値を求め、さらに各リンクを除くことによる目的関数の増加の下限値を計算し直すのである。しかし、各リンクの目的関数の増加の下限値 f_k の再評価を行えば、 f_k / l_k の順となっているリンクの順序も決め直さなくてはならない。そこで、ここでは f_k の再評価は行わない。そのかわり、ある計算段階のネットワークを最大ネットワークのようにみなして目的関数値を求め直す操作を頻繁に行うことにする。この操作を行うと、それより前に $x_k = 0$ と固定したリンクについては式（3．7）の補助問題の目的関数の計算に入れず、下限値を求める式（3．10）では Z^0 のかわりに求め直した目的関数値を使うことになる。

3．5．3　計算に用いる例題

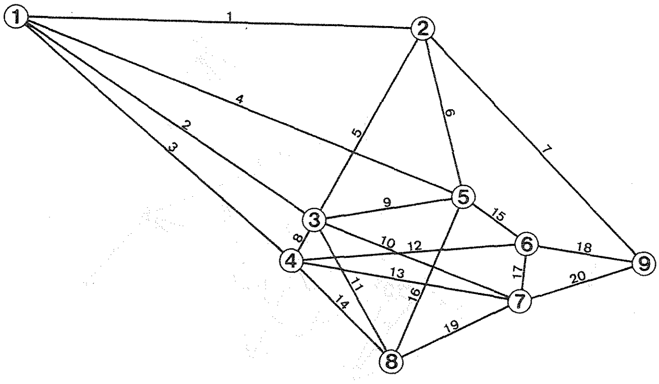
ここでは5種類のネットワークを例題に用いる。そして、それぞれの例題においてネットワーク長の制約値を6種類与え、合計30個の問題を解く。



図—3.7　例題3.1の最大ネットワーク



図—3.8．例題3.2の最大ネットワーク



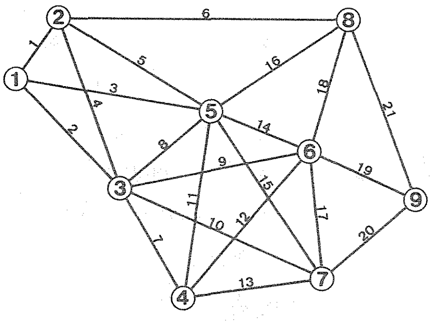
図—3.9　例題3.3の最大ネットワーク

例題３．１～３．５の最大ネットワークを図－３．７～３．11に示す。そしてリンク長をまとめて表－３．２に示す。例題３．１～３．３は Scott と Hoang が用いているネットワークであるが、例題３．２、３．３のリンクの長さは10倍にしている。例題３．４、３．５はノードの位置を乱数を使って決め、隣接するノード間をリンクでつないだものである。これらの例題では、リンク長はノード間の空間距離に比例している。ノード数とリンク数からすれば、ネットワークの規模は例題３．１がもっとも小さく、例題３．２、３．３、３．４、３．５の順に大きくなる。

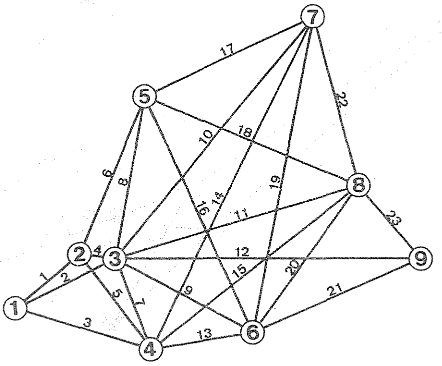
表—3.2　例題のリンクの長さ

リンク	リンク長 例題3.1	例題3.2	例題3.3	例題3.4	例題3.5
1	39	350	540	199	222
2	32	260	480	401	296
3	33	235	490	526	377
4	49	180	640	483	97
5	27	300	290	477	318
6	43	320	230	772	455
7	36	200	430	338	260
8	44	360	60	318	440
9	20	230	200	514	417
10	24	380	290	590	835
11	35	210	210	502	678
12	19	340	310	514	816
13	21	430	310	373	275
14	14	360	190	283	988
15	21	380	105	535	705
16	10	120	240	438	694
17		230	75	340	498
18		260	160	363	618
19			190	313	860
20			170	325	483
21				511	489
22					468
23					258

ネットワーク長の制約値は最短生成木の長さ以上で、最大ネットワークの長さ以下でなければ



図— 3.10　例題 3. 4 の最大ネットワーク



図— 3.11　例題 3. 5 の最大ネットワーク

ならない。そこで、最短生成木の長さを０％、最大ネットワークの長さを100％としてネットワーク長の制約値の水準を表わす。そして、制約水準を10％、25％、40％、55％、70％、85％としてネットワーク長の制約値を与えることにする。それぞれの例題でのネットワーク長の制約値をまとめたのが表－３．３である。

表— 3. 3　例題のネットワーク長の制約値

制約水準(％)	ネットワーク長の制約値				
	例題 3. 1	例題 3. 2	例題 3. 3	例題 3. 4	例題 3. 5
10	157	1779	1902	3198	3407
25	208	2340	2520	4184	4764
40	260	2901	3138	5170	6121
55	312	3462	3756	6156	7477
70	364	4023	4374	7143	8834
85	415	4584	4992	8129	10190

３．５．４　計算結果と考察

問題を解くにあたり、目的関数の下限値を強める操作はリンクを３本除くごとに行うことにする。最大ネットワークからリンクを１本除いたときの目的関数値は初期解を求めるときに計算しているのでこれを利用する。そして、以後は $x_k \leftarrow 0$ とした除去リンク数が４、７、10、……となるたびに、それらのリンクを除いたネットワークの目的関数値を求める。

例題の計算には神戸大学計算センターの ACOS700S を使用し、計算時間は表－３．４のとおりであった。計算過程で目的関数値を調べた実行可能解の数は表－３．５に示すとおりである。図－３．12、３．13は計算時間の比較を容易にするためのものである。図－３．12は、ネットワーク長の制約水準が85％のときの計算時間に対する各制約水準での計算時間の比率を表わしている。例題ごとに比率を求め、それらを相乗平均したものである。図－３．13は、アルゴリズムＥ１の計算時間に対するアルゴリズムＥ２～Ｅ４の計算時間の比率を表わしたものである。

これらの図表から、次のことがわかる。

① 例題３．１から例題３．５へとネットワークの規模が大きくなるほどアルゴリズム間の差

表—3.4　例題を解くのに要した計算時間

例　　題	制約水準（%）	計算時間（ms） アルゴリズムＥ１	アルゴリズムＥ２	アルゴリズムＥ３	アルゴリズムＥ４
例題3.1	10	338	192	386	412
	25	157	142	227	233
	40	94	99	93	107
	55	60	65	58	57
	70	65	66	59	59
	85	56	60	55	54
例題3.2	10	2698	2248	1089	1176
	25	795	460	505	555
	40	1199	619	634	532
	55	283	201	255	183
	70	126	138	114	129
	85	80	89	78	86
例題3.3	10	6148	1731	1751	1845
	25	4445	581	1098	837
	40	2961	622	1366	780
	55	327	235	312	259
	70	292	198	256	217
	85	144	140	135	134
例題3.4	10	30529	22901	14253	14511
	25	3305	1252	3090	2935
	40	4726	1937	2896	2479
	55	1673	712	919	693
	70	400	472	367	442
	85	152	143	136	131
例題3.5	10	29038	27312	13202	14656
	25	12965	12150	6657	10408
	40	1455	707	1449	1179
	55	912	730	783	785
	70	297	260	259	241
	85	141	148	131	134

が顕著になる。

② ネットワーク長の制約値が小さいほどアルゴリズム間の差が大きい。

③ ネットワーク長の制約値が小さくなれば、計算時間は急激に増加する。

④ ネットワーク長の制約水準が10%、70%、85%のときはアルゴリズムＥ３の計算時間がもっとも短く、25%と40%のときにはアルゴリズムＥ２の計算時間が最短である。

⑤ 計算過程で目的関数値を調べた実行可能解の数と計算時間とは比例していない。

⑥ 実行可能解の数を比べると、アルゴリズムＥ１よりもアルゴリズムＥ２は多く、アルゴリズムＥ３よりもアルゴリズムＥ４が多い。また、アルゴリズムＥ３、Ｅ４はそれぞれアルゴ

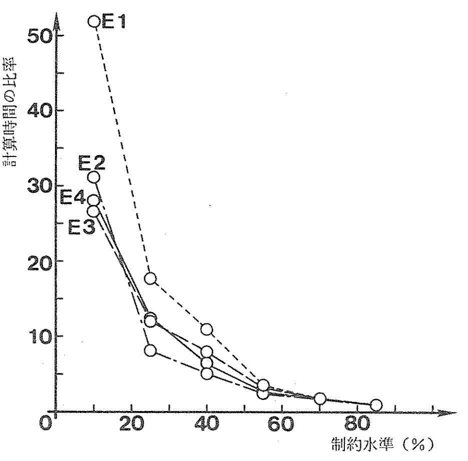
表―3.5　例題の計算過程で目的関数値を調べた実行可能解の数

例　　題	制約水準（%）	実行可能解の数 アルゴリズムE1	アルゴリズムE2	アルゴリズムE3	アルゴリズムE4
例題3.1	10	2	14	2	6
	25	3	12	3	11
	40	2	10	2	7
	55	2	2	2	2
	70	5	5	5	5
	85	4	4	4	4
例題3.2	10	19	36	15	19
	25	1	3	1	3
	40	91	134	64	77
	55	3	2	3	2
	70	6	10	6	10
	85	2	5	2	5
例題3.3	10	182	21	5	8
	25	43	22	12	9
	40	173	74	66	60
	55	8	2	8	7
	70	37	12	35	19
	85	4	3	4	3
例題3.4	10	208	1930	94	213
	25	2	132	2	28
	40	6	16	6	11
	55	41	17	39	17
	70	60	78	60	78
	85	8	7	8	7
例題3.5	10	29	429	21	169
	25	16	194	11	112
	40	1	13	1	8
	55	6	67	6	53
	70	10	7	10	7
	85	1	2	1	2

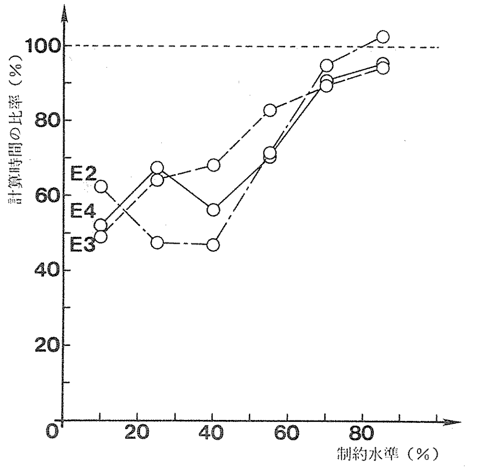
リズムE1、E2よりも少ない。

以下ではアルゴリズムE1～E4を単にE1～E4で表わす。

E1とE3の計算時間を比べると、例題3.1のネットワーク長の制約水準が10%と25%の場合を除いてE3の方が短い。ところが図－3.13を見れば、E3の計算時間の短縮量は制約水準が10%、25%のときには大きい。このふたつの場合だけがそれ以外と逆になっているのは、例題3.1のネットワーク規模が小さく、複雑な分枝方法が適さないからだと考えられる。このことは、例題を解いてアルゴリズムを比較するとき、適切な例題とネットワーク長の制約水準を選ばなければ誤った結論を導く可能性があることを示している。計算時間が問題となるのはネットワー



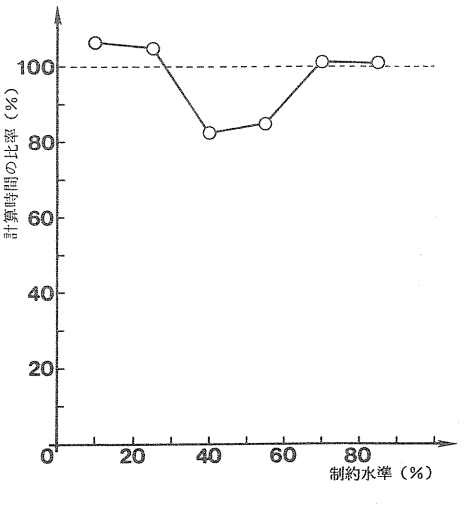
図―3.12　ネットワーク長の制約水準と計算時間の関係



図―3.13　アルゴリズムＥ１の計算時間に対する比率

ク規模が大きく、ネットワーク長の制約値が小さい場合である。アルゴリズムの改良の効果をみるには、このような場合の計算時間を中心に検討しなければならない。

まず実行可能化分枝を用いることの効果を検討する。このためにはＥ１とＥ２、Ｅ３とＥ４を比較すればよい。図－３．13でＥ１とＥ２を比べると、ネットワーク長の制約水準が40%のときを最大にして、Ｅ２の計算時間の短縮量は大きい。Ｅ３とＥ４の比較のためにＥ３の計算時間に対するＥ４の計算時間の比率を表わしたのが図－３．14である。この場合も制約水準が40%と55%のときは計算時間はＥ４が短いが、それ以外では逆にＥ４の方が少し長い。



図―3.14　アルゴリズムＥ３の計算時間に対するアルゴリズムＥ４の計算時間の比率

表－３．５で計算過程において目的関数値を調べた実行可能解の数を見ると、実行可能化分枝を用いるＥ２、Ｅ４の方がＥ１、Ｅ３よりも多い傾向にある。これは実行可能化分枝の考え方から予想されたことである。実行可能化分枝の方が実行可能解が少ない場合は、計算時間の短縮量がかなり大きい。

これらをまとめると、実行可能化分枝を用いることの効果は次のようであったといえる。

① 実行可能化分枝を用いれば、計算過程で多くの実行可能解の目的関数値を調べなければならないことが多いが、計算時間は短い。

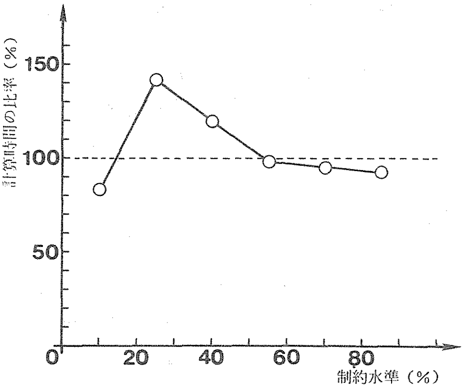
② 実行可能化分枝を用いることによる計算時間の短縮量は、あるネットワーク長の制約水準のときに最大となる。それよりも制約がゆるくてもきつくても、短縮量は小さくなる傾向にある。

③ 連結網化分枝を用いるならば、実行可能化分枝を使っても計算時間はあまり短くならず、

逆に少し長くなることも多い。

連結網化分枝を用いることの効果を検討するにはE1とE3、E2とE4を比べればよい。図－3.13からは、連結網化分枝を用いることにより、E3の計算時間はE1よりもかなり短くなり、しかもネットワーク長の制約がきつくなるほど短縮量が大きいことがわかる。ネットワーク長の制約値が小さいときには非連結網である実行可能解が増えるが、E3ではそれらの解が探索範囲から除かれるために計算時間が短くなるのである。このことは、表－3.5において連結網化分枝を用いた場合に目的関数値を調べる実行可能解の数が少なく、とくにネットワーク長の制約値が小さいときに少ないことで裏づけられる。

図－3.15でE2とE4とを比べると、制約水準が25%と40%のときにはE4の方が計算時間がかなり長い。これは、連結網化分枝が最適解ないしはそれに近い解に早く到達するのに適していないことを示している。非連結網である解を除外する効果が実行可能化分枝より非効率であることによる計算時間の増加を下回っているのである。これに対し、実行可能化分枝と併用しないときは、連結網化分枝の採用でどの制約水準でも計算時間が短くなっている。これは、連結網化分枝が実行可能化分枝と似た役割を果たしたからであろう。すなわち、一度に複数の分枝変数の値を固定するため、分枝操作が簡単になったからだと考えられる。



図―3.15　アルゴリズムE2の計算時間に対するアルゴリズムE4の計算時間の比率

図－3.12ではネットワーク長の制約水準が低くなれば、どのアルゴリズムでも計算時間が急激に長くなっている。しかし、E1とE2に比べると、E3とE4の増加はゆるやかである。これも、非連結網である解を探索範囲から除いた効果を示している。ここでは、ネットワーク長の制約水準が10%以上の問題を解いたが、さらに制約をきつくすれば連結網化分枝を用いる効果が大きく現われると考えられる。そこで、例題3.3で制約水準が2%の問題を解くと、E1～E4の計算時間は、それぞれ8713ms、2467ms、762ms、816msであった。このように、E1とE2では計算時間はなお急激に増加し、連結網化分枝を用いたE3とE4では逆に短くなっている。

連結網化分枝を用いた効果をまとめると、次のようになる。

① 実行可能化分枝を用いない場合、連結網化分枝を使えば計算時間が短くなる。そして、ネットワーク長の制約値が小さいほど計算時間の短縮量が大きい。

② 実行可能化分枝を用いる場合、連結網化分枝を使っても計算時間を短縮する効果は小さく、逆に計算時間が増加することもある。しかし、計算時間が長くなる傾向にあるネットワーク長の制約値が小さいときに、計算時間を短縮する効果が比較的大きい。

③ ネットワーク長の制約値が小さくなると、連結網化分枝を使わなければ計算時間が急激に増加する。連結網化分枝を用いれば、あるピークを境に計算時間が短くなる。

これらのアルゴリズムからどれを採用するかを決めるには、例題のすべての問題を解くのに要

した計算時間がひとつの指標になる。例題３．１～３．５の30個の問題を解くのに要した計算時間を求め、Ｅ１を100％とすると、Ｅ２～Ｅ４の計算時間は、それぞれ72.4％、49.7％、53.1％である。これから判断すれば、アルゴリズムＥ３が最短の計算時間で、アルゴリズムＥ４がそれよりも少し長いということになる。すなわち、Hoang の分枝方法に連結網化分枝を組み合わせたアルゴリズムが、ここで用いた例題に対してはもっとも良かったということである。

３．５．５　むすび

ここでは、例題を解いてアルゴリズムを比較し、実行可能化分枝と連結網化分枝を用いたときの計算時間の短縮の効果を調べた。例題の計算結果をもとにした分枝方法に関する考察をまとめると、次のようになる。

① 連結網化分枝を含め、一度に複数の分枝変数の値を固定する分枝方法を使って分枝操作を簡単にすれば、計算時間が短くなる。

② 連結網化分枝を用いれば、ネットワーク長の制約値が小さい場合でも短い計算時間で解が得られる。

３．６　結　　語

この章の目的は厳密解法のアルゴリズムの改良であった。従来のアルゴリズムを検討し、その結果をもとにして新しい分枝方法を用いるアルゴリズムを提案した。

従来のアルゴリズムの検討の結果として、次のことを得た。

① 分枝限定法と分枝後退法を比べると、分枝限定法の必要な記憶容量が大きいという短所は重大である。そのために近似解法への応用も困難である。

② Scott のアルゴリズムは分枝後退法の基礎的なものであるが、計算効率が悪い。Hoang は目的関数の下限値を使うことにより、計算効率を改善している。

③ アルゴリズムを改良する方法には、一度に複数の分枝変数の値を固定する分枝操作の採用と、ネットワークが非連結網になる解を作らない分枝方法の採用とが考えられる。

そして、実行可能解まで一度に分枝する実行可能化分枝と、連結網である解だけからなる解集合へ分枝する連結網化分枝を提案した。実行可能化分枝を使うか Hoang の分枝方法を使うか、連結網化分枝を用いるかどうかで４種類のアルゴリズムを作成し、例題を解いて比較した。その結果は次のとおりであった。

① 一度に複数の分枝変数の値を固定する分枝方法を用いて分枝操作を簡単にすれば、計算時間が短くなる。

② 連結網化分枝を用いれば、ネットワーク長の制約値の小さい場合でも短い計算時間で解が得られる。

ここで提案した分枝方法は、計算時間を短縮することに十分な効果があった。とくに連結網化分枝は、それを使わない場合には計算時間が極めて長くなる問題に対して有効であった。連結網化分枝は最適交通網構成問題の特性を利用したものである。アルゴリズムを改良するには、通常の０－１整数計画問題の解法に近いものを対象問題に適した解法にすることが必要であることが示されたといえよう。

このように、厳密解法のアルゴリズムを改良するという目的を達することができた。ここでは最適交通網構成問題の基本問題が対象であった。しかし、他の問題に対しても、ここで行った改良の手順が応用できるであろう。

参考文献

1) Steenbrink, P.A. : Optimization of Transport Networks, John Wijey & Sons, 1974.

2) Scott, A. J. : The optimal network problem : some computational procedures, Transportation Research, Vol. 3 , pp.201〜210, 1969.

3) Hoang, H. H. : A computational approach to the selection of an optimal network, Management Science, Vol. 15, №5 , pp. 488〜498, January, 1973.

4) 枝村俊郎・森津秀夫：最適交通ネットワーク問題の厳密解法と近似解法、土木学会論文報告集、第262号、pp.113〜127、1977年6月.

第４章　最適交通網構成問題の近似解法に関する考察

４．１　概　　説

この章では、基本的な最適交通網構成問題を対象とし、近似解法についての考察を行う。ここでの目的は、短い計算時間で最適解ないしはそれに近い解を求めることができる近似解法の作成である。まず従来の近似解法を検討する。そして、その検討結果を生かし、解の精度と計算時間の両方の改善を図り、厳密解法を応用した近似解法を提案する。さらに、大規模な問題を解くことのできる近似解法を提案する。

４．２では、従来の近似解法の代表的なものとして、最短生成木法、Scott の近似解法、飯田の近似解法、Dionne らの近似解法を検討する。つぎに、その結果からどのように新しい近似解法を作ればよいかを考える。

４．３では、厳密解法を応用した２種類の近似解法を提案する。それは、厳密解法を繰り返し適用してリンクの段階的削減を行うものと、backward 法の解の近傍において厳密解法を使った局所最適化を行うものである。

４．４では、大規模な最適交通網構成問題を解くのに適した近似解法を提案する。これは、簡単なリンク評価値を使って forward 法または backward 法の手順で解を求めるものである。そして、これらの解法を厳密解法と組み合わせたものも提案する。

４．５では、提案した近似解法と従来の近似解法を比較する。例題を解いて必要な計算時間と解の精度を調べ、その結果を考察する。

４．６では、この章で得られた成果をまとめる。

４．２　従来の近似解法の検討

４．２．１　はじめに

この節では、最適交通網構成問題の近似解法の主要なものを取り上げて検討する。そして、従来の近似解法では不十分な点を補う新しい近似解法を作成する際の方針を導く。

この章では厳密解法の考察と同様に、問題３．１の最適交通網構成問題の基本問題を対象とする。この節で検討する近似解法のほとんども、この問題を扱ったものである。

４．２．２では、最適交通網構成問題の近似解法のなかで、もっとも簡単な最短生成木法を検討する。

４．２．３では、最適交通網構成問題の代表的な近似解法である Scott の近似解法を検討する。

４．２．４では、飯田のふたつの近似解法を検討する。backward 法を改良したものと、動的計画法を応用したものである。

４．２．５では、厳密解法のアルゴリズムを修正して作成した Dionne らの近似解法を検討する。

４．２．６では、従来の近似解法の検討結果をもとに、新しい近似解法はどのように作らなければならないか述べる。

4.2.2　最短生成木法

最短生成木法[1)]は、もっとも簡単な最適交通網構成問題の近似解法である。最初に最大ネットワークの最短生成木を求め、制約の範囲内で長さの短いリンクを順に付加してゆく方法である。求めた解の目的関数値が必要なければ、最短経路探索を行わなくてよい。そのため、大規模な問題でも手作業で解くことができる。目的関数値が必要なときでも、解のネットワークに対してだけ最短経路を探せばよい。したがって、この場合でも必要な計算時間は短い。

最短生成木法では、リンクの長さをリンクの重要度を表わす評価値にしていると考えることができる。目的関数値が有限であるためには解が連結網でなければならない。そこで、評価値である長さの順にリンクを採択して連結網を構成し、さらにリンクを付加する手順になる。リンクの長さはネットワークの状態によって変化しないので、評価値を計算し直す必要はない。長さの短いリンクは最短経路に含まれることが多く、最適ネットワークに含まれることも多いと考えられる。よって、この解法は目的関数を直接には扱っていないが、得られる近似解は最短距離の合計の最小化に反するものではないであろう。

最短生成木法の欠点のひとつは、ネットワークの状態すなわちリンク相互の関係に応じてリンクの評価値が変わらないことである。連結網になるまでは、長いリンクが先に選ばれることもあるが、それ以後は必ずリンク長の順に採択される。そのため、目的関数値の改善の効果の小さいリンクが多数選ばれ、それらよりは長くても目的関数値の改善の効果の大きいリンクが採択されない場合が起きる。

目的関数が直接に計算に関係しないことが、もうひとつの欠点である。目的関数値を計算しないことは、計算時間の短縮に極めて有効である。しかし、目的関数が違う問題に対しても同じ解しか求めようがない。そのため、ここで対象としている基本問題の場合にはよいが、拡張した最適交通網構成問題への適用は困難である。たとえば、目的関数においてノード間の最短距離に対する重みがそれぞれ異なる場合に、重みを考慮した解法に直すことは容易でない。この問題は需要交通量を最短経路に配分し、総走行距離を最小にするネットワークを求めることに相当する。解法が単純な構造であるため、基本的な考え方をそのままにして応用できないのである。

最短生成木法は長さの短い順にリンクを最短生成木につけ加えてゆく方法である。これとは逆に、最大ネットワークから長い順にリンクを除いてゆく方法が考えられる。このようにしても、得られる解は最短生成木法と同じである。しかし、計算機を使用する場合、次の点で最短生成木法に劣る。

① ネットワークの連結性の検査は、最短生成木法では生成木ができるまで行えばよい。逆の手順では、リンクを除いても非連結にならないかをつねに調べなければならない。

② 最短生成木法での連結性の検査は、付加しようとするリンクの両端のノードがすでにつながれていないかを調べればよい。これは、互いに連結されたノードに同一のラベルをつけてゆくことで簡単に行える。逆の手順を用いる場合には、除こうとするリンクがなくても両端のノードが連結されているかを調べなければならない。ノードが孤立する場合は容易にわかるが複数のノードからなる連結成分に分かれる場合は調べにくい。

したがって、最大ネットワークから少しのリンクを除けば解が求まるとき以外は、最短生成木

法を使う方が計算時間が短いと考えられる。よって、最大ネットワークから長い順にリンクを除いてゆく方法は、あらためて検討しなくてよい。

最短生成木法は最適交通網構成問題の基本問題を対象とするもっとも簡単な近似解法である。そして、手作業で解が得られる唯一の解法といってよい。しかし、ここでの考察の結果、需要交通量を考えた問題への適用が困難であり、実用的な価値に乏しいことがわかった。

4.2.3 Scott の近似解法

Scott は、forward solution algorithm と backward solution algorithm の2種類の近似解法を提案している[2)]。これらは目的関数値の変化量を基準にし、ネットワークを構成するリンクを1本ずつ増加または減少させる方法である。最適交通網構成問題の近似解法の研究では、これらの解法との計算時間や解の精度の比較がよく行われる。

forward solution algorithm は最短生成木から出発し、目的関数値の最大の減少をもたらすリンクをネットワークにつけ加える操作を繰り返す方法である。ここでは、これを Scott の forward 法とよぶことにする。backward solution algorithm は最大ネットワークから出発し、目的関数値の増加が最小ですむリンクをネットワークから除く操作を繰り返す方法である。ここでは、これを Scott の backward 法とよぶことにする。

Scott のふたつの近似解法の特徴はネットワーク変換操作を行うことである。これは、計算の各段階でネットワークに含まれているリンクと含まれていないリンクを1本ずつ入れ換え、近傍での最適化を図るものである。これを導入したのは、この操作を行わない場合には解の精度が悪いからである。つまり、近似解と最適解との目的関数値の差が大きいのである。

Scott の近似解法の基本的な部分は、ネットワーク変換操作を除いた部分である。一般には、この基本的な部分を forward 法、backward 法とよんでいることが多い。ここでも、それに従う。ネットワーク変換操作を行わないこれらの解法もよく使われ、最適交通網構成問題の近似解法の基礎になっている。そこで、これらの計算手順を示しておく。

forward 法の計算手順は次のとおりである。

ステップ1：最短生成木を求め、それを初期ネットワークとする。

ステップ2：ネットワークに含まれず、ネットワークにつけ加えてもネットワーク長が制約値を超えないリンクについて、それぞれをネットワークに加えたときの目的関数値の減少量を求める。つけ加えることのできるリンクがなければ、計算を終える。

ステップ3：目的関数値の減少量が最大のリンクをネットワークにつけ加える。ステップ2へもどる。

backward 法の計算手順は次のとおりである。

ステップ1：最大ネットワークを初期ネットワークとする。

ステップ2：ネットワークを構成するリンクについて、それぞれを除いたときの目的関数値の増加量を求める。

ステップ3：目的関数値の増加量が最小のリンクをネットワークから除く。ただし、どのリンクを除いても非連結網になり、目的関数値が無限大になるならば、実行可能解は求まらないが

計算を終える。

ステップ４：ネットワーク長が制約値を超えているならば、ステップ２へもどる。制約値以下ならばステップ５へ進む。

ステップ５：ネットワークに含まれず、ネットワークにつけ加えてもネットワーク長が制約値を超えないリンクについて、それぞれをネットワークに加えたときの目的関数値の減少量を求める。つけ加えることのできるリンクがなければ、計算を終える。

ステップ６：目的関数値の減少量が最大のリンクをネットワークにつけ加える。ステップ５へもどる。

forward 法はネットワークを実行可能な状態に保って目的関数値を改善してゆく方法であり、主実行可能な方法に相当する。backward 法は目的関数値が最良の状態から出発して実行可能になるようにしてゆく方法で、双対実行可能な方法に相当する。

この計算手順で明らかなように、forward 法と backward 法では、ネットワークへの付加あるいはネットワークからの除去を決められたリンクは、その決定を再検討されることはない。backward 法で実行可能解に達したとき、すでに除いたリンクのなかに、再びネットワークに加えることのできる短いリンクがないか探されるだけである。Scott がネットワーク変換操作を用いているのは、この欠点を補うためであると考えられる。ネットワーク変換操作は forward 法のステップ１とステップ３のあと、backward 法のステップ３とステップ６のあとで行われる。なお Scott の backward 法は完全連結網を初期ネットワークにしているが、解法を検討する際には重要なことではない。

Scott の近似解法を評価するには、ネットワーク変換操作の効果を検討しなければならない。ネットワーク変換操作を行うことでどれだけ解の精度が向上したか、Scott は具体的な数字を示していない。しかし、ネットワーク変換操作で解が改良されることは明らかである。問題になるのは、この操作に必要な計算量が目的関数値の改善に見合うかどうかである。

いま、最大ネットワークのリンク数がmで、３本のリンクを除いた解を backward 法で求めるとする。ネットワーク変換操作を行わなければ、$m+(m-1)+(m-2)$回の目的関数値の計算で解が求まる。これに対し、ネットワーク変換操作はそれだけで、$2(m-2)+3(m-3)$回程度の目的関数値の計算が必要である。この操作でリンクを入れ換えたならば、回数はさらに大きくなる。このように、ネットワーク変換操作に必要な計算量は backward 法の基本的な部分の計算量よりもはるかに大きい。ネットワーク変換操作はネットワークの近傍をすべて調べつくそうとするものであり、一種の厳密解法を使っていると解釈できる。しかも、その探索は系統だった効率的なものではない。したがって、対象とするネットワークの規模が大きくなれば、計算時間が問題になるであろう。

最短生成木法は拡張した問題への適用が困難なことが欠点であった。だが forward 法と backward 法では目的関数値の変化量を考えており、拡張した問題へも容易に適用できる。適用が困難なのは、ネットワークからリンクを除けば目的関数値が改善されることがある場合や、実行可能領域が複雑な場合に限られるであろう。

Scott の近似解法についての考察をまとめると、次のようになる。

① ネットワーク変換操作は計算効率が悪く、解法の基本的な部分よりも計算量が大きい。この操作で解は改良されるが、大規模なネットワークを扱う場合には計算時間が問題になると予想できる。

② forward 法と backward 法の考え方は拡張した最適交通網構成問題へも適用できる。

4.2.4 飯田の近似解法

飯田は最適道路網構成問題の2種類の近似解法を提案している[3)]。対象とする問題は、建設費の制約下で需要交通量を最短経路配分したときの総走行距離を最小化するものである。解法を考えるうえからは、この問題はここで対象としている最適交通網構成問題の基本問題と大きな違いはない。これらの解法は基本問題にそのまま適用できる。

飯田の近似解法Ⅰは、最大ネットワークからリンクを1本除いたものを初期ネットワークとし、backward 法をリンク数だけ繰り返す方法である。backward 法では局所的最適に落ち込むことが多いので、初期ネットワークを多数与えることにより、これを解決しようとしたものである。

この解法は backward 法を完全に含んでいるため、得られる解はそれと同じか、あるいは改良されたものである。しかし、初期ネットワークを変えればつねに異なる解が得られるわけではない。その場合はむだな計算を繰り返すことになる。解を求めるのに要する計算時間は、最大ネットワークのリンク数を m とすると、backward 法のほぼ m 倍と考えられる。ネットワークにおけるノード数とリンク数の関係からすれば、ノード数が10～20位のネットワークでも backward 法の20倍から200倍ほどの計算時間になる。この近似解法で必要な計算時間はかなり長いといえる。

飯田の近似解法Ⅱ（DP的探索法）は動的計画法（Dinamic Programming : DP）を応用したものである。最適交通網構成問題では最適性原理が成り立たない。そのため、動的計画法で最適解を求めることはできない。しかし、得られる解が最適解である保証がいらない近似解法には用いることができる。計算手順は backward 法と同じように最大ネットワークから出発し、ネットワークを構成するリンクを減らしてゆくものである。backward 法との違いは、各段階において、どのリンクを除いてその段階へ進んだかによって、リンク数だけのネットワークを保持することである。これにより、1段階進むと除去リンクの組み合わせが大幅に変わる可能性があり、最適解に近い解が得られる可能性があるとしている。さらに、除去リンクの可能な組み合わせの数がほとんど一定であり、計算時間が長くならないことも特徴にあげている。

近似解法Ⅱは backward 法を包含しているわけではないが、組み合わせを多く調べるため、最適解に近い解が得られる可能性が高いことは確かであろう。しかし、必要な計算時間はやはり長いと考えられる。最大ネットワークのリンク数が m で、3本のリンクを除いた解を求めるとすると、$m+m(m-1)+m(m-1)$ 回程度の目的関数値の計算が必要である。これは backward 法のほぼ m 倍であり、近似解法Ⅰと大差ない。そして、backward 法での目的関数値の計算回数が最大ネットワークのリンク数に比例するとすれば、近似解法Ⅱではリンク数の2乗に比例することになる。ネットワークの規模が大きくなれば、近似解法Ⅱでの計算時間は急激に長くなること

が予想できる。さらに、近似解法Ⅱの計算過程では同一のネットワークを複数保持する可能性がある。その場合はむだな計算を繰り返すことになる。

飯田の２種類の近似解法は、どちらも backward 法よりは得られる解の精度が高いと考えられる。しかし、むだな計算を繰り返す可能性があり、必要な計算時間は長い。

4.2.5 Dionne・Florian の近似解法

Dionne と Florian は Hoang の分枝後退法をもとにした近似解法を提案している[1)]。これは、厳密解法のアルゴリズムをそのまま使い、最適解を除く可能性があっても分枝をできるだけ早い段階で打ち切り、計算時間を短縮しようとしたものである。厳密解法を基礎にしていることが特徴である。

Dionne らは、まず Hoang の分枝後退法で使われている最短経路探索のアルゴリズムを他のものに代えている。しかし、これは最適交通網構成問題の解法における本質的なことではない。つぎに、目的関数の下限値を求めるときに超過見積りをすることによって近似解法に直している。Hoang の方法での目的関数の下限値は、リンクを１本除くことにより生じる目的関数値の増加の下限値を合計して得られる。この値は最大ネットワークからリンクを１本除き、そのリンクの両端のノード間の最短距離を求め、これとリンク長の差で計算できる。Dionne らは、リンクを１本除いたネットワークと最大ネットワークの目的関数値の差をかわりに用いているのである。こうすれば下限値ではなくなるため、最適解を探索範囲から除いてしまうこともある。けれども探索範囲がせまくなり、計算時間が短くなる。

Dionne らは、この近似解法を用いた計算例ではほとんどの場合に最適解が得られ、計算時間も短いとしている。Hoang の提案している目的関数の下限値は強力でない。そのため、超過見積りによって良い結果が得られているのであろう。しかし、アルゴリズムは厳密解法とまったく同じであることから、厳密解法と同じ欠点を持つと指摘している。すなわち、規模の大きい問題でネットワーク長の制約値が小さい場合に計算時間が非常に長くなることである。彼らの場合は、第３章で提案した連結網化分枝を用いていないのであるから、このようなときの計算時間の増加は極めて大きいであろう。そして、連結網化分枝を用いたとしても、厳密解法と同じ構造であるために規模の大きい問題には適用できないと考えられる。

Dionne らの近似解法は厳密解法とほとんど同じであるため、得られる解は最適解に近く、むだな計算を繰り返すこともないと考えられる。しかし、厳密解法の欠点を引き継いでおり、大規模な問題になれば計算時間が急激に増加することが予想される。

4.2.6 新しい近似解法の作成方法

ここでは、従来の近似解法の検討結果をもとに、それらの欠点を補った新しい近似解法を作成するにはどうすればよいかを述べる。

最適交通網構成問題の基本問題に対する近似解法のなかで、もっとも簡単なものは最短生成木法である。しかし、最短生成木法は得られる解の精度が低く、応用の可能性もほとんどない。そのため、近似解法の基本になっているのは forward 法と backward 法である。従来の近似解法

の研究は、このふたつの解法が出発点になっているといえる。

Scott 自身は forward 法と backward 法にネットワーク変換操作を導入して近傍のネットワークに対する最適化を図り、解を改良しようとしている。飯田の近似解法は backward 法の考え方を基礎とし、計算途中の段階で多くのネットワークを保持することにより、最適解に近い解が得られる可能性を高めようとするものである。Dionne らの近似解法はこれらと異なり、厳密解法での分枝を途中で打ち切り、計算時間を大幅に短縮しようとするものである。したがって、これは forward 法などを基礎とするものではないが、初期解は backward 法で求めている。そして、解の精度や計算時間を backward 法と比較している。このように、これらの近似解法は単純な backward 法などよりも解の精度を高めることを目的に作られたものである。改良された解が得られることは解法の構成から明らかであるか、計算例で確められている。

近似解法は実際の交通網計画に必要な規模の問題を解くことを前提とし、実用性を重視したものである。近似解法を作成する際の目標は次の2点である。

① 最適解ないしはそれに近い解が得られること。

② 解を求めるのに必要な計算時間が短いこと。

Scott、飯田、Dionne らの近似解法は、計算時間を短くすることよりも、解の精度を高めることに重点を置いて作られている。しかし、規模の大きい問題を扱わなければならないのであるから、むだな計算を省いて計算効率を高めることにも配慮しなければならない。Scott と飯田の近似解法は、この点において改良を加える必要がある。

Dionne らは厳密解法のアルゴリズムを応用しているが、これが計算効率を高めるひとつの方法である。厳密解法のアルゴリズムは計算時間を短くする改良がなされており、解の探索を系統的に行うからである。しかし、Dionne らの近似解法は厳密解法のアルゴリズムそのままであり、計算時間について十分に検討されているとはいえない。

厳密解法の近似解法への応用方法に関しては、3.2.2で解の探索範囲を限定する方法と他の近似解法に組み込んで解の精度を向上させる方法があることを述べた。厳密解法の特性を考慮したうえで応用方法を考えれば、精度の高い解を短い計算時間で得られる近似解法ができるであろう。

これまでの近似解法の研究では、それぞれの思いつきをもとに解法が組み立てられていた。厳密解法の研究では多くの例題が解かれているが、その成果を生かす試みはみられない。近似解法は得られる解が最適解であるという保証は必要なく、理論的な根拠が必ずしも明確でなくてもよい。そこで、厳密解法で例題を解くという実験の結果を使い、最適解ないしはそれに近い解を短い計算時間で求める方法を考えることができるはずである。

実際の交通網計画では大規模なネットワークを扱わなければならないことがある。このような場合、解の精度に重点を置いて作られた近似解法の使用は困難である。ネットワークが大きくなったときに計算時間が増加するのは、調べなければならない解の数が増えることと、目的関数値の計算に要する計算時間の増加が原因である。これらが、どの程度の増え方かを考えてみる。

最適交通網構成問題は目的関数に最短経路問題を含んでいる。最短経路を求めるための計算量は、ネットワークのノード数を　として n^2 あるいは n^3 に比例するとされている[4)]。そして、

最短生成木法以外ではもっとも簡単な解法である forward 法や backward 法でも、解を得るのに必要な目的関数値の計算回数は、リンク数を m として m ないし m^2 に比例する。リンク数とノード数との関係は、完全連結網では $m=n(n-1)/2$ であり、リンク数はノード数の２乗に比例する。したがって、backward 法などの計算時間は n^4 ないし n^5 に比例すると考えられる。よって、backward 法でさえ解くことのできる問題の規模は大きくない。

最短生成木法は最短経路問題を含む目的関数値の計算をしなくてもよいため、大規模な問題を解くことができる。しかし、解の精度と応用の可能性の点で実用的でないことは、これまでに述べてきたとおりである。そこで、forward 法や backward 法よりも短い計算時間で最短生成木法よりも良い解が得られ、拡張した問題への応用が可能な近似解法の開発が必要である。その場合、最短経路の探索回数が少なくてすむようにしなければならない。

以上の検討の結果、次の２点に基づいて近似解法を作成することにする。

① 厳密解法で例題の最適解を求め、その考察結果を利用して最適解ないしはそれに近い解を短い計算時間で求められる近似解法を作成する。このとき、厳密解法の応用を考える。

② 大規模問題を解くために、backward 法などよりも短い計算時間で最短生成木法よりも良い解を与える近似解法を作成する。このとき、最短経路探索が少なくてすむようにする。

４．２．７　むすび

この節では従来の近似解法の主要なものを取り上げ、解の精度と計算時間について考察した。ただし、ここで実際に例題を解いたのではなく、解法の構成と発表されている計算結果から考察した。そして、それをもとに新しい近似解法をどのように作ればよいかを検討した。

従来の近似解法は、最短生成木法と Scott、飯田、それに Dionne らの近似解法を調べた。これらの近似解法は、最短生成木法を除いて解の精度を高めることを目標に作られ、計算時間の増加への配慮が不十分であった。なかには、むだな計算を繰り返す可能性のある近似解法もあった。いずれも大規模な問題への適用が困難であると予想された。

これらのことから、精度の高い解を短い計算時間で求められるものと、大規模な問題へ適用できるものとの２種類の近似解法を作成することにした。そのために、厳密解法を用いて例題を解き、最適解の特性を調べることを考えた。そして、計算効率を高めるために厳密解法を応用した近似解法を作成することにした。もう一方の大規模問題へ適用するためのものは、最短経路探索の回数を減らし、最短生成木法と backward 法との中間的な性格をもつ近似解法にする。

４．３　厳密解法を応用した近似解法の提案[5)]

４．３．１　はじめに

この節の目的は、最適解ないしはそれに近い解を短い計算時間で求められる近似解法を作成することである。まず厳密解法でいくつかの例題を解き、どのようにすれば最適解が得られる可能性の高い近似解法ができるかを検討する。そして、その検討結果をもとに、厳密解法を応用したふたつの近似解法を提案する。それは、段階的削減を行うものと局所最適化を行うものである。

４．３．２では、厳密解法でいくつかの例題の最適解を求め、最適解の特徴を調べる。そして、

精度の高い解を効率良く求める方法を考える。

4．3．3では、厳密解法を繰り返し適用し、段階的削減を行う近似解法を提案する。

4．3．4では、backward 法の解の近傍において、厳密解法を用いて局所最適化を行う近似解法を提案する。

4．3．2　最適解の検討

ここでは、最適交通網構成問題の最適解がどのようなものであるかを検討する。多数の問題を解き、それぞれの最適ネットワークを構成するリンクに共通の特徴を調べる。そして、それをもとに近似解法を改良するにはどうすればよいかを考える。

最適解の検討には第3章で計算例に用いた例題3．1～3．5を使用する。最適ネットワークの変化を詳しくみるために、制約水準は5％から95％の範囲で5％きざみで与える。すなわち、各例題ごとに19個ずつの合計95個の問題を解く。各問題のネットワークの総延長の制約値は表－4．1に示すとおりである。

表—4.1　例題のネットワーク長の制約値

制約水準	ネットワーク長の制約値				
（％）	例題3.1	例題3.2	例題3.3	例題3.4	例題3.5
5	139	1592	1696	2869	2955
10	157	1779	1902	3198	3407
15	174	1966	2108	3526	3860
20	191	2153	2314	3855	4312
25	208	2340	2520	4184	4764
30	226	2527	2726	4513	5216
35	243	2714	2932	4841	5668
40	260	2901	3138	5170	6121
45	277	3088	3344	5499	6573
50	295	3275	3550	5828	7025
55	312	3462	3756	6156	7477
60	329	3649	3962	6485	7929
65	346	3836	4168	6814	8382
70	364	4023	4374	7143	8834
75	381	4210	4580	7471	9286
80	398	4397	4786	7800	9738
85	415	4584	4992	8129	10190
90	433	4771	5198	8458	10643
95	450	4958	5404	8786	11095

これらの問題を解いた結果、表－4．2～4．6に示す最適解が得られた。これらの表は、例題ごとにそれぞれの制約水準の最適ネットワークにどのリンクが含まれるかを表わしている。代表的な近似解法である backward 法の近似解と比べやすいように、リンクの順序は backward 法でネットワークから除かれる順にしてある。ただし、backward 法で最後に残る生成木を構成するリンクはリンク長の大きい順にしている。そして、表中の横線はそこまでのリンクを除いたとき、

表—4.2　例題 3.1 の最適解

リンク	制約水準（%）																		
	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
8																			
4																		○	○
11													○	○	○	○	○	○	○
6															○	○	○	○	○
15				○		○		○			○	○	○	○	○	○	○	○	○
10			○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
1									○	○	○	○	○	○	○	○	○	○	○
2							○	○	○	○	○	○	○	○	○	○	○	○	○
7					○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
14		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
5	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
13	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
9	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
12	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
16	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

表—4.3　例題 3.2 の最適解

リンク	制約水準（%）																		
	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
13																			
18										○		○		○		○	○	○	○
1																		○	○
8													○	○	○	○	○	○	○
15															○	○	○	○	○
10											○	○	○	○	○	○	○	○	○
6									○	○	○	○	○	○	○	○	○	○	○
12		○					○	○	○	○	○	○	○	○	○	○	○	○	○
5					○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3				○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
14			○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
9	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
17	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
11	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
7	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
16	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

表—4.4　例題 3.3 の最適解

リンク	制約水準（%）																		
	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
11				○			○				○	○	○	○	○	○	○	○	○
13													○			○		○	
7														○			○	○	○
3																			○
12								○		○	○	○	○	○	○	○	○	○	○
4															○	○	○	○	○
20		○				○	○	○	○	○	○	○	○	○	○	○	○	○	○
16					○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
10				○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
1									○	○	○	○	○	○	○	○	○	○	○
5			○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
14	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
6	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
9	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
19	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
18	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
15	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
17	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

表—4.5　例題 3.4 の最適解

リンク	制約水準（%）																		
	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
6																			
15																○		○	○
11																○	○	○	○
21															○	○	○	○	○
9													○	○	○	○	○	○	○
4								○				○	○	○	○	○	○	○	○
10										○	○	○	○	○	○		○	○	○
12								○	○	○	○	○	○	○	○	○	○	○	○
3							○		○	○	○	○	○	○	○	○	○	○	○
20					○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
16				○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
13		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
5	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2			○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
18	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
17	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
7		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
19	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
14	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

表—4.6　例題 3.5 の最適解

リンク	制約水準（%）																		
	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
5															○		○		○
15																			
19																		○	○
2											○	○		○	○	○	○	○	○
14																○	○	○	○
16											○	○	○	○	○	○	○	○	○
6					○			○		○		○	○	○	○	○	○	○	○
12													○	○	○	○	○	○	○
3		○		○	○		○	○	○	○	○	○	○	○	○	○	○	○	○
10									○	○	○	○	○	○	○	○	○	○	○
9	○			○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
21				○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
18						○	○	○	○	○	○	○	○	○	○	○	○	○	○
11			○			○	○	○	○	○	○	○	○	○	○	○	○	○	○
17		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
20	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
22	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
13	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
7	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
23	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

backward 法では実行可能解に達することを表わしている。

これらの表から次のことがわかる。

① 最適ネットワークは、それよりも制約水準のゆるい問題の最適ネットワークの部分ネットワークであることが多い。

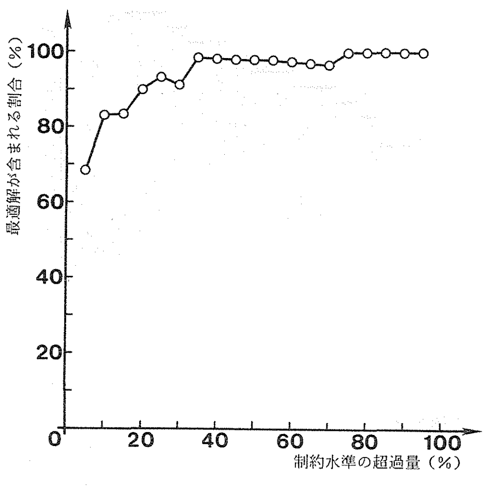
② リンクを除いてゆくだけの backward 法でも、得られる解が最適解であることが多い。

③ リンクを除いてゆくだけの backward 法では最適解が得られないとき、それまでに除かれているリンクを付加すれば最適解になる場合がいくらかある。

④ backward 法の解が最適解でないとき、近似解が得られる前後で除かれるいくらかのリンクの組み合わせを調べれば、最適解が求まることが多い。

⑤ backward 法で得られる生成木を構成するリンクは、ほとんどの問題の最適ネットワークに含まれている。

①は、ある問題を解くとき、それよりもネットワーク長の制約値が大きい問題の最適ネットワークを最大ネットワークとみなし、そこに含まれるリンクの組み合わせだけを調べても最適解が求まる場合が多いことを示している。そこで、制約水準の大きい問題の最適ネットワークにどれだけの割合で最適ネットワークが含まれているかを調べると、図－４．１が得られた。この図を見れば、たとえば制約水準が20％大きい問題の最適ネットワークを構成するリクの組み合わせから解を探すと、これらの問題のうちの90％の問題で最適解が得られることがわかる。制約水準の超過量を大きくすれば、最適解が得られる割合は増加する。ただし、超過量をさらに大きくしても増加量は徐々に減少してゆく。

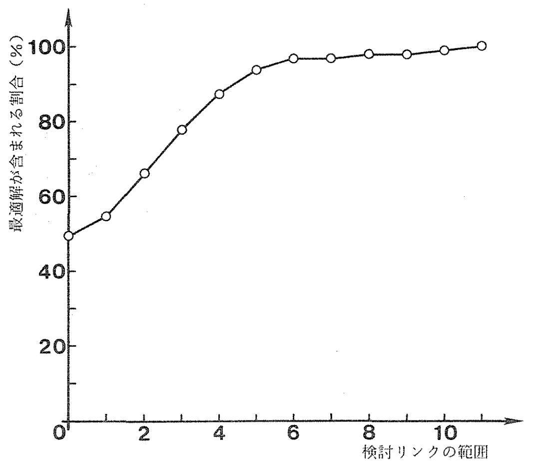


図－4.1　制約水準の超過量と最適解が含まれる割合の関係

②について調べると、リンクを除いてゆくだけの backward 法によって、95問題のうちの47問題で最適解が求まることがわかる。backward 法ではネットワーク長が制約値以下になったとき、forward 法の手順でそれまでに除いたリンクのなかにネットワークにつけ加えることができるものがないかを調べる。これを行えば、③の場合の最適解が得られる。③に相当するのは19問題で、そのうちの２問題では２本のリンクを付加し、残りは１本のリンクを付加すれば最適解になる。１本のリンクをつけ加えるだけでよい場合は、forward 法の手順で最適解が求まる。したがって、一般的な backward 法ではここでの95問題のうちの64問題、つまり67％の問題で最適解が得られることになる。簡単な backward 法では良い解が得られないとされているが、最適解でないのは⅓の問題だけである。

④は backward 法の解の近傍を探せば最適解が得られる可能性が高いことを表わしている。そ

こで、backward 法の解の近傍にどれだけの割合で最適解があるかを調べる。その結果は図－４．２に示すとおりである。図の横軸の検討リンクの範囲は、backward 法の解の近傍のどれだけの範囲を探すかを表わすものである。すなわち、リンクの順序を表－４．２～４．６と同じにし、最大ネットワークから順にリンクを除くものとする。最大ネットワークのリンク数を m とし、リンク $1 \sim r$ の r 本のリンクを除いたときにネットワーク長が制約値以下になったとする。検討リンクの範囲が M ならば、リンク r の前後それぞれ M 本のリンクを検討リンクとし、残りのリンクは解の状態で固定する。そして、検討リンクの組み合わせを調べたときに、どれだけの割合で最適解が得られるかを表わしたのが図－４．２である。検討リンクの範囲を M とすれば最大２ M 本のリンクが検討リンクになるが、 $r<M$ のときや、 $m-r<M$ のときの検討リンク数はこれよりも小さい。



図—4.2　検討リンクの範囲と最適解が得られる割合の関係

図－４．２を見れば、検討リンクの範囲と最適解が得られる割合の関係は対数曲線状になっている。検討リンクの範囲を拡大すれば最適解が得られる割合は高くなるが、さらに範囲を広げてもその変化量は小さくなってゆく。この図では、たとえば検討リンクの範囲を５、すなわち最大10本のリンクを検討リンクにすれば、95問題の94％で最適解が得られることがわかる。

このように、backward 法の解の近傍を探せば最適解が得られることが多い。しかし、他の近似解法ではさらに近くに最適解があるかもしれない。そこで、最短生成木法、forward 法、backward 法で決定されるリンクの順序のうち、どれを使えば近似解の近くに最適解があるかを調べる。そのために、それぞれの問題で近似解から最適解がどれだけ離れているかを式（４．１）で表わす。

$$s=\sum_{k=1}^{r}(r+1-k)^2 x_k^* + \sum_{k=r+1}^{m}(k-r)^2(x_k^*-1) \qquad (4.1)$$

ここに、 s ：近似解と最適解の隔たりを表わす指標

r ：最大ネットワークからリンク $1 \sim r$ を除けば近似解になるとする。

x_k^*: 最適解におけるリンク k の状態を表わす。

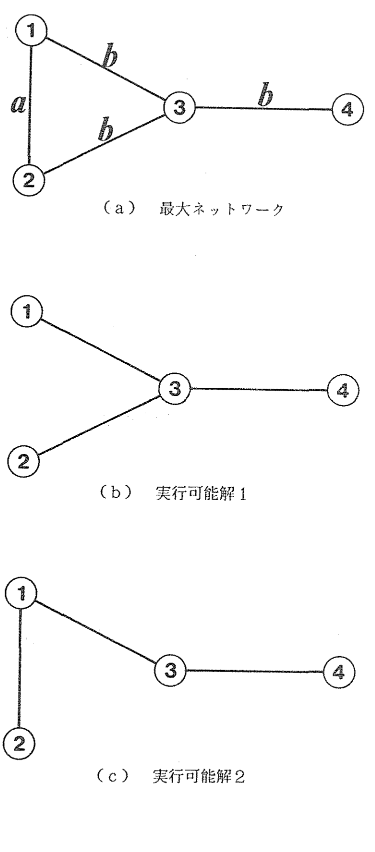
リンクの順序は backward 法で除かれる順、forward 法でネットワークにつけ加えられる順を逆にしたもの、最短生成木法でネットワークにつけ加えられる順を逆にしたものとする。そして、式（４．１）で求めた指標を95問題で平均する。

計算の結果、指標の平均値は backward 法の順序で10.9、forward 法の順序で11.7、最短生成木法の順序で34.6であった。forward 法との差は小さいが、backward 法の順序による場合が近

似解と最適解の隔たりがもっとも小さい。

⑤については、backward 法で得られる生成木が最適ネットワークに含まれていないのは例題3.4の2問題だけである。解のネットワークに含めるかどうかを検討するリンクを減らすことは、計算時間の短縮に大きな効果がある。⑤の結果はこれに利用できそうである。そこで、この問題をさらに検討する。

最適解に必ず含まれるリンクがないかは、これまでも考えられてきた。しかし、ここでの計算例でもわかるように、どの最適解にも必ず含まれるリンクの集合はありそうにない。計算例を見れば、長さが最短のリンクは必ず最適解に含まれると思える。だが、これは図－4.3に示す例で否定される。



（a） 最大ネットワーク

（b） 実行可能解1

（c） 実行可能解2

図—4.3 最短のリンクが最適解に含まれない例

図－4.3（a）のネットワークにおいて、3本のリンクの長さは b、1本のリンクの長さは a で、$b>a$ とする。ネットワーク長の制約値が $3b$ よりも大きいときは、図－4.3（b）、（c）の実行可能解が最適解の候補となる。ノード1、3の間のリンクをはずした解は実行可能解2と同じ目的関数値である。実行可能解1の目的関数値は $9b$ で、実行可能解2の目的関数値は $3a+7b$ である。そこで、$a>2/3b$ のときはリンク長が最短のものを含まない実行可能解1が最適解である。

このように、必ず最適解に含まれるリンクを見つけることは容易でなく、そのようなリンクが存在するかもはっきりしない。けれどもここでの計算例でみるかぎり、backward 法で得られる生成木を構成するリンクはそれに近いものである。厳密解法では使えなくても、近似解法であれば、これらのリンクを解のネットワークにつねに含めるようにしてもよい。

以上の最適解についての検討の結果から、最適解ないしはそれに近い解を短い計算時間で求められる近似解法の作成には、次の点を考慮することが必要であると考える。

① ある問題を解くとき、それよりも制約水準の大きい問題の最適ネットワークを構成するリンクの組み合わせを調べれば、最適解が得られることが多い。制約水準の超過量を大きくすれば、最適解が得られる可能性が大きくなる。

② リンクを除いてゆくだけの backward 法でも最適解が得られることが多い。そうでないと

きも、その近似解の近傍に最適解があることがほとんどである。

③ backward 法で得られる生成木を構成するリンクは、ほとんどの場合の最適ネットワークに含まれている。

4.3.3 段階的削減を行う近似解法の提案

最適解を調べると、ある問題の最適解がそれよりも制約水準の大きい問題の最適ネットワークを構成するリンクの組み合わせで得られることが多かった。ここではこれを利用し、厳密解法を繰り返し使ってリンクの段階的削減を行う近似解法を提案する。

リンクの最適な組み合わせを求めるには厳密解法を用いなければならない。これは、最大ネットワークの全リンクを調べる場合も、ある最適ネットワークに含まれるリンクを調べる場合も同じである。しかし、制約水準が大きく最大ネットワークから除くリンクが少ないときは、厳密解法でも計算時間が非常に短かった。このことと最適解の検討結果から、厳密解法を繰り返し用いる近似解法が考えられる。すなわち、ネットワーク長の制約水準を順次小さくして厳密解法を繰り返し適用し、最適ネットワークから次の最適ネットワークを探す方法である。このようにすれば、最適解が得られる可能性が高く、しかも全体としての計算時間は短いことが期待できる。

この考え方は backward 法と似ている。しかし、backward 法ではリンクの長さに関係なくつねに1本ずつのリンクを除くのに対し、一般に複数のリンクからなるある長さのリンク集合を除くことが異なる。そのため、解のネットワークに早い段階で除かれたリンクを付加しても、なお実行可能であるという状態は起こりにくい。

ここで提案する近似解法は、backward 法のリンクを除く計算過程を厳密解法に変更し、複数のリンクを同時に除くようにしたものといえる。すなわち、厳密解法を近似解法に組み合わせて解の精度を高めようとするものである。一度に除くリンクを多くしなければ、厳密解法の部分での計算時間は長くならない。しかし、少しでも計算時間を短くするために解の探索範囲を限定する。4.3.2では、リンクを除いてゆくだけの backward 法の解の近傍に最適解があることが多いと述べた。よって、厳密解法を適用するとき、ネットワークから除くかどうかを調べるリンクは backward 法の解の近傍のものだけとする。

段階的削減を行う近似解法の基本的な考え方は以上のとおりである。実際に計算するときは、ネットワーク長の制約値を小さくしてゆく幅を決めなければならない。この幅が大きければ最適解が得られる可能性が高くなるが、一度に除くリンクが多くなるので計算時間は長くなる。そこで、ネットワーク長の制約値の減少幅はパラメータとして与え、解の精度と計算時間を制御できるようにする。このパラメータが与えられたとき、繰り返し計算の各回のネットワーク長の制約値は次のように決める。

$$N = max\left\{\left[\frac{L^{max} - L^{C}}{\delta} + 0.5\right], 1\right\} \qquad (4.2)$$

$$L_h^C = L^{max} - \left(\frac{L^{max} - L^C}{N}\right) h \quad (h = 1, 2, \cdots, N) \qquad (4.3)$$

ここに、N ：厳密解法を用いる繰り返し計算回数

L^{max}：最大ネットワークの総延長

L^C：ネットワーク長の制約値

δ ：ネットワーク長の制約値の減少幅のパラメータ

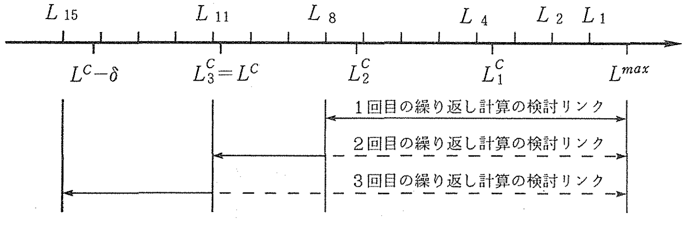
L^C_h：h 回目の繰り返し計算のときのネットワーク長の制約値

ただし、式（4．2）の［　］はガウス記号である。

式（4．2）は繰り返し計算回数を決めるものである。除かなければならないリンクの合計長を制約値の減少幅のパラメータで割り、四捨五入している。そして、繰り返し計算回数は少なくとも1回とすることを表わしている。式（4．3）は、繰り返し計算において、ほぼ等しい合計長のリンク集合を除くように各回の制約値を決めるものである。

繰り返し計算で厳密解法を適用するときの検討リンクは次のように決める。まずリンクをbackward 法でネットワークから除かれる順に並べる。backward 法で残る生成木を構成するリンクはリンク長の長い順とする。そして、検討リンクは順番に除いたとき、次の繰り返し計算の問題の実行可能解になる最小の範囲とする。検討リンクよりあとの順番のリンクは、その回の計算においては解のネットワークに含めるように固定する。

検討リンクの決定方法を図－4．4の例によって説明する。リンクを backward 法での順に並べ、ネットワーク長を数直線上に表わすと図－4．4になるとする。繰り返し計算は3回とする。backward 法で除かれる順に k 番目までのリンクを最大ネットワークから除いたときのネットワーク長を L_k で表わす。この場合、リンク8までを除けば2回目の問題の実行可能解になる。そこで、1回目の計算での検討リンクはリンク1～8とする。同じようにして、2回目の検討リンクはリンク1～8のうちで1回目の計算の最適解に含まれるリンクとリンク9～11である。そして、3回目の検討リンクはリンク1～11のうちで2回目の最適解に含まれるリンクとリンク12～15である。



図－4.4　段階的削減を行う近似解法での検討リンク

この方法ではネットワーク長の制約値の減少幅を2倍し、それを平均リンク長で割れば、ほぼ厳密解法を適用するときの検討リンク数になる。ネットワーク長の制約値の減少幅のパラメータの値を変えれば検討リンクの範囲も変わる。これにより、解の精度と計算時間のパラメータ値による変化はいっそう大きくなる。

段階的削減を行う近似解法のアルゴリズムは次のようになる。

ステップ1：繰り返し計算回数 N と繰り返し計算を行うときのネットワーク長の制約値 L^C_h を計算する。

ステップ2：ネットワーク長が $L^C-\delta$ 以下になるまで backward 法の計算を行い、繰り返し計算を行うときの検討リンクと初期解を求める。

ステップ3：$h \leftarrow 1$ とする。

ステップ4：h 回目の繰り返し計算の検討リンクと固定リンクを定める。

ステップ5：厳密解法を用い、ネットワーク長が L_h^C 以下の最適解を検討リンクの組み合わせから探す。

ステップ6：$h \leftarrow h+1$ とし、$h \leqq N$ ならばステップ4へもどる。$h > N$ ならば計算を終える。

最後の繰り返し計算で求めた解が、この解法による近似解である。ステップ5では厳密解法を用いるが、そのアルゴリズムには**3.5**の計算例で計算時間が短かったアルゴリズムE3を使用する。これは連結網化分枝を行い、連結になってからは Hoang の分枝方法を用いるアルゴリズムである。

ステップ2では backward 法の計算を行うが、この部分では次の関係を利用してむだな計算を省く。すなわち、あるネットワークを $\boldsymbol{x}$ とし、$\boldsymbol{x}'$ は $\boldsymbol{x}$ からリンクを除いた部分ネットワークとする。いま、それぞれのネットワークからリンク k を除いたときの目的関数値を $Z_k(\boldsymbol{x})$, $Z_k(\boldsymbol{x}')$ とする。そうすれば、これらの間には式（4.4）の関係が成り立つ。

$$Z_k(\boldsymbol{x}') \geqq Z_k(\boldsymbol{x}) \tag{4.4}$$

ネットワーク $\boldsymbol{x}'$ からリンクを1本除いて目的関数値を調べてゆき、ある計算段階ではリンク j を除いたネットワークの目的関数値が最小であるとする。このとき、$Z_k(\boldsymbol{x}) \geqq Z_j(\boldsymbol{x}')$ ならばリンク k をネットワーク $\boldsymbol{x}'$ から除いて目的関数値を計算する必要はない。$Z_k(\boldsymbol{x}) < Z_j(\boldsymbol{x}')$ のときにだけ目的関数値を計算して比較を行えばよい。これは、$Z_j(\boldsymbol{x}')$, $Z_k(\boldsymbol{x})$, $Z_k(\boldsymbol{x}')$ の間の関係は次のどれかだからである。

i) $Z_k(\boldsymbol{x}) \geqq Z_j(\boldsymbol{x}')$ のとき

$Z_k(\boldsymbol{x}') \geqq Z_k(\boldsymbol{x}) \geqq Z_j(\boldsymbol{x}')$

ii) $Z_k(\boldsymbol{x}) < Z_j(\boldsymbol{x}')$ のとき

a) $Z_k(\boldsymbol{x}) \leqq Z_k(\boldsymbol{x}') < Z_j(\boldsymbol{x}')$

b) $Z_k(\boldsymbol{x}) < Z_j(\boldsymbol{x}') \leqq Z_k(\boldsymbol{x}')$

ネットワークからリンクを1本除いて目的関数値を調べるとき、一般にリンク番号順に調べてゆく。ここに述べた関係を利用して目的関数値の計算を省略するのならば、リンクを除いたときに目的関数値が小さいものが先に調べられる方がよい。そこで、そのような状態に近づけるために、最大ネットワークからリンクを1本除いて目的関数値を計算したとき、その値の小さい順にリンクを並べ換えておく。また、除くと非連結網になるリンクがあれば、それ以後はそのリンクを除く操作を行わない。

4.3.2では、backward 法で得られる生成木を構成するリンクはほとんどの問題の最適解に含まれると述べた。しかし、この近似解法ではそれらのリンクを解のネットワークに含めるように固定していない。それは、厳密解法に連結網化分枝を行うアルゴリズムを用いるからである。連結網化分枝を行えば、解のネットワークが生成木かそれに近いときには計算時間が短かった。そのため、それらのリンクを固定して計算時間の短縮を図るより、良い解が得られる可能性を高くする方がよい。

段階的削減を行う近似解法の特徴と改良点をまとめると、次のようになる。

① 厳密解法を使い、いくらかの長さのリンク集合をネットワークから除く操作を繰り返す。

② 厳密解法で繰り返し計算を行うときのネットワーク長の制約値の減少幅を変えることにより、解の精度と必要な計算時間を変えることができる。

③ 厳密解法を適用するとき、backward 法の計算結果を利用して検討リンクを限定する。このため、厳密解法で調べるリンクの組み合わせは限られる。

④ 厳密解法の初期解を求め、検討リンクを限定するために backward 法を用いる。backward 法の部分ではむだな目的関数値の計算を省略している。

①がこの近似解法の最大の特徴であり、解法の発想そのものである。複数のリンクをまとめて除くことを想定しているが、一度に除くリンクの合計長はネットワーク長の制約値の減少幅のパラメータで変わる。そして、このパラメータの値を変えれば、それに応じて検討リンクの範囲も変わるのである。

パラメータの値を大きくすれば一度に除くリンクの数が大きくなり、検討リンク数も大きくなる。そのため計算時間は長くなるが、最適解やそれに近い解が求まる可能性は高くなる。パラメータの値を極端に大きくすれば、繰り返し計算は1回だけで、すべてのリンクが検討リンクになる。これは完全な厳密解法による計算である。逆にパラメータの値を小さくすれば、一度に除くリンク数が小さくなるので厳密解法での計算量が減る。検討リンク数も減るので計算時間が短くなると考えられるが、解の精度は悪くなる。そして、パラメータ値を極端に小さくすればリンクを1本ずつ除くことになり、リンクを除いてゆくだけの backward 法と同じ解を与える。このように、パラメータ値を変えれば、この解法は簡単な backward 法にも厳密解法にもなるのである。

対象とするネットワークの規模が大きくなったときの計算時間の変化は、backward 法のようにリンク数との関係を明確にできない。backward 法を含んでいるのであるから、backward 法よりも計算時間が長いことは明らかである。厳密解法の部分ではパラメータ値によって検討リンク数が決まり、それはネットワークが大きくなっても変化しない。したがって、厳密解法の部分での計算時間は、ネットワーク規模の変化によってそれほど急激には変化しないと考えられる。パラメータの値を一定にすると、規模の小さいネットワークに対しては厳密解法となり、規模の大きいネットワークに対しては簡単な近似解法になるといえる。

4.3.4 局所最適化を行う近似解法の提案

最適解を調べると、リンクを除いてゆくだけの backward 法でも最適解が得られることが多く、そうでないときも近似解の近傍に最適解があることがほとんどであった。ここではこれを利用し、backward 法の解の近傍を厳密解法で探索する近似解法を提案する。

ここでの近傍とは、backward 法で実行可能解が得られる前後にネットワークから除かれるリンクの組み合わせでできる範囲である。つまり、これらのリンクを検討リンクとし、その最適な組み合わせを求めるのである。このためには厳密解法を使わなければならない。しかし、この場合の探索範囲は backward 法の近傍だけであるから、厳密解法でも計算時間は短い。

backward 法の解の近傍で最適化を図ることは、Scott の backward 法と似ている。Scott の backward 法はリンクを1本除くごとにネットワーク変換操作を行い、近傍のネットワークから最適なものを選ぶ。ネットワーク変換操作を実行可能解が求まったときにだけ行うようにすれば、ここで提案する解法に近くなる。しかし、Scott の解法でのネットワーク変換操作の計算効率が悪いことは、4.2.3で述べたとおりである。厳密解法を用いる方が系統的な探索を行うために計算時間が短く、確実に最適な組み合わせを選び出すことができる。

ここで提案する解法では、次のようにして検討リンクを設定する。まずリンクを backward 法

でネットワークから除かれる順に並べ換える。backward 法で残る生成木を構成するリンクはリンク長の長い順とする。そして、ネットワーク長の制約に対し、最大ネットワークからリンク $1\sim r$ を除いたときに実行可能になるとする。このとき、検討リンクの範囲を決めるパラメータを M とし、リンクを以下に示す3種類に分ける。

① リンク $1\sim(r-M)$

② リンク $(r-M+1)\sim(r+M)$

③ リンク $(r+M+1)\sim m$

①のリンクは解のネットワークに含めない。②のリンクは解のネットワークに含めるかどうかを検討する。③のリンクは解のネットワークに含める。ただし、$r\leqq M$ のときは解のネットワークに含めないとするリンクはなく、$r+M\geqq m$ のときは解のネットワークに含めるリンクはない。検討リンクはbackward 法の解に含まれるリンクと含まれないリンクから、それぞれ M 本以内が採用される。このパラメータの値を変えれば解の探索範囲が変わり、それに応じて解の精度と計算時間も変わることになる。

厳密解法を用い、backward 法の解の近傍で局所最適化を行う近似解法のアルゴリズムは次のようになる。

ステップ1：backward 法で初期解を求める。そして、さらにネットワークからリンクを M 本除くまで backward 法の計算を続ける。

ステップ2：backward 法での計算結果によってリンクを分類し、検討リンクと固定リンクを定める。

ステップ3：厳密解法によって検討リンクの最適な組み合わせを求める。

厳密解法には連結網化分枝を行うアルゴリズムE3を用いる。そして、backward 法の部分では段階的削減を行う近似解法と同じようにしてむだな目的関数値の計算を省略する。

局所最適化を行う近似解法の特徴と改良点をまとめると、次のようになる。

① 厳密解法によって backward 法の解の近傍を探索する。

② 厳密解法で組み合わせを調べる検討リンクの範囲を変えることにより、解の精度と必要な計算時間を変えられる。

③ 厳密解法の初期解を求め、検討リンクを限定するために backward 法を用いる。backward 法の部分ではむだな目的関数値の計算を省略している。

検討リンクの範囲を決めるパラメータの値を大きくすれば、最適解ないしはそれに近い解が得られる可能性が高くなる。しかし、必要な計算時間は長くなる。パラメータ値を極端に大きくすればすべてのリンクが検討リンクになり、この解法は厳密解法そのものになる。逆にパラメータ値を小さくすれば解の探索範囲が狭くなり、解の精度は悪くなるが計算時間は短くなる。とくにパラメータの値を0として検討リンクを設けない場合、この解法はリンクを除いてゆくだけの backward 法になる。この近似解法も段階的削減を行う近似解法と同じように、パラメータの値を変えれば簡単な backward 法にも厳密解法にもなるのである。段階的削減を行う近似解法と異なるのは、パラメータ値を大きくすれば得られる解は同じか改良されるかのどちらかであることが明らかな点である。

対象とするネットワークの規模が大きくなった場合、計算時間の変化は backward 法よりは急

激だが厳密解法よりはゆるやかである。そして、パラメータ値を一定にすると、ネットワークの規模が小さいときには厳密解法になり、ネットワークの規模が大きくなれば相対的に backward 法に近づく。

4.3.5 むすび

この節では、最適解ないしはそれに近い解を短い計算時間で求められることを目標に近似解法を作成した。まず、いくつかの例題の最適解を調べ、短い計算時間で精度の高い解を得る方法を考えた。そして、それに基づいた２種類の近似解法を作った。

最初に95問題の最適解を検討した。その結果、最適ネットワークが制約水準の大きい問題の最適ネットワークの部分ネットワークであることが多かった。そして、リンクを除いてゆくだけの backward 法でも最適解が得られることが多く、そうでないときも近似解の近傍に最適解があることがほとんどであった。そこで、これらを利用した近似解法を作成した。

段階的削減を行う近似解法は、ネットワーク長の制約水準を順次小さくして厳密解法を繰り返し適用し、最適ネットワークから次の最適ネットワークを探すものである。局所最適化を行う近似解法は、backward 法で実行可能解が得られる前後にネットワークから除かれるリンクを検討リンクとし、厳密解法でその最適な組み合わせを探すものである。

これらの近似解法は厳密解法を応用し、パラメータの値によって簡単な backward 法にも厳密解法にもなることが特徴である。

4.4 大規模問題を扱うための近似解法の提案[6)]

4.4.1 はじめに

この節では、大規模な最適交通網構成問題を解くのに適した近似解法を提案する。forward 法や backward 法よりも計算時間が短く、最短生成木法よりも解の精度と応用の可能性が優れた近似解法の作成が目的である。まず計算時間を大幅に短縮する方法を考え、ネットワークにおけるそれぞれのリンクの重要度を表わす評価値を検討する。そして、このリンク評価値を用いた近似解法を提案する。

4.4.2では、大規模問題を解くために用いることを前提としたリンク評価値を提案する。

4.4.3では、リンク評価値を使って forward 法の手順で解を求める簡易 forward 法を提案する。

4.4.4では、リンク評価値を使って backward 法の手順で解を求める簡易 backward 法を提案する。

4.4.5では、簡易 backward 法を用いてリンクの段階的削減を行う近似解法と、簡易 forward 法の解の近傍で局所最適化を行う近似解法を提案する。

4.4.2 リンク評価値の検討

ここでは、最適交通網構成問題の近似解法で使われるリンクの評価値について検討する。そして、大規模な問題を扱うのに適するリンク評価値を提案する。

最適交通網構成問題の規模が大きくなったときに計算時間が急激に長くなる原因は、目的関数の計算回数の増加と、目的関数値の計算に含まれる最短経路探索に要する計算時間の増加である。このうち、最短経路探索の計算時間は解法の改良では短くならない。最適交通網構成問題の解法の改良でできることは、目的関数値の計算回数を減らすことである。したがって、forward 法や backward 法よりも計算時間の短い近似解法を作るには、これらより目的関数値の計算回数が少なくてすむようにしなければならない。

forward 法や backward 法、それに最短生成木法の特徴は、ネットワークに含めるかあるいはネットワークから除くリンクを１本ずつ決めてゆくことである。こうすれば、実行可能解を作って比較する方法よりも目的関数値の計算回数は少ない。そして、最短生成木法は付加リンクの決定に目的関数値を必要としないため、求めた近似解の目的関数値を計算するだけでよい。この点が forward 法や backward 法と異なる。forward 法、backward 法より目的関数値の計算回数を減らすには、付加リンクや除去リンクの決定の際の目的関数値の計算を減らすことを考えなければならない。

最短生成木法では付加リンクを決めるとき、リンクの長さがネットワークにおけるリンクの重要度の指標にされている。すなわちリンク長を評価値とし、評価値の順にリンクをネットワークに付け加えているのである。しかし、この方法は目的関数を直接には扱っていないため、解の精度と応用の可能性に問題があった。一方、forward 法や backward 法ではリンクの付加や除去による目的関数値の変化量をリンクの評価値にしている。大規模問題に適用でき、最短生成木法の短所を補った近似解法にするには、これらの中間的な性格を持つリンク評価値を考えなければならない。

最短経路探索を繰り返さないのならば、ネットワークが変化するたびに目的関数値を計算したり、最短経路を求め直すことはできない。そこで、最大ネットワークで求めた値を使ってリンク評価値を作成する。それぞれのリンクがネットワークで果たしている機能を考え、そのリンクが除かれた場合を想定してリンクの評価値を作る。

最大ネットワークにおけるノード $i\ j$ 間の最短距離を d_{ij}^1 、第２最短距離すなわち第２最短経路の長さを d_{ij}^2 とする。そうすれば、最大ネットワークでのノード $i\ j$ 間の最短経路を構成するリンクのどれか１本でも除かれると、ノード ij 間の最短距離は d_{ij}^2 以上になる。このことからノード ij 間の最短経路は $(d_{ij}^2-d_{ij}^1)$ の価値を持つと考えることができる。対象としている最適交通網構成問題の制約条件はネットワークの総延長である。よって、最短経路としての価値をリンクごとに集計し、単位長あたりの値にすれば、式（４．５）になる。

$$f_k'=\frac{1}{l_k}\left\{\sum_i \sum_j \delta_{ijk}(d_{ij}^2-d_{ij}^1)\right\} \qquad (4.5)$$

ここに、f_k' ：リンク k の最短経路としての評価値

l_k ：リンク k の長さ

δ_{ijk} ：リンク k が最大ネットワークにおけるノード ij 間の最短経路に含まれるとき１、含まれないとき０とする。

たとえば、リンク k の両端のノード間の第２最短距離が最短距離である l_k に近いとき、最短

経路としての評価値 f_k' は０に近い。そして最短経路がノード間の唯一の経路である場合は、最短経路を構成するリンクでは $f_k'=\infty$ となる。このように、式（４．５）で求まる f_k' が大きいほど、最短経路としての重要なリンクであるとする。

除かれたリンクの影響を評価値に反映させるには、ネットワークが変わるたびに最短距離と第２最短距離を求め直せばよい。しかし、それでは backward 法などよりは最短経路探索の回数は減るが、なお除去リンク数と同じ数の最短経路探索が必要である。そして、最大ネットワークで計算した評価値の順序で解を求める方法では、解の精度を最短生成木法からあまり向上させられないであろう。そこで、最大ネットワーク以外では最短経路探索を行わないで、しかもネットワークの変化を評価値に反映させる方法を考えなければならない。

あるリンクがネットワークから除かれると、これまでそのリンクが果たしていた機能を他のリンクが受け持つことになる。そこで、ネットワークに含まれないリンクがあるとき、そのリンクの両端のノード間における第２最短経路が機能を代替すると仮定する。しかし、機能を完全に代替することはできないであろう。よって、機能を代替できる割合は第２最短距離とリンク長との差が小さければ１に近く、差が大きくなれば０に近づくようにする。そして、代替できる機能は除かれたリンクが最大ネットワークで最短経路として果たしていたものだけとし、式（４．５）で表わされているとする。このような仮定のもとに第２最短経路としての評価を加え、式（４．５）と式（４．６）でリンクの評価値を定める。

$$f_k = f_k' + \sum_h (1 - x_h)\gamma_{kh} f_h' \frac{l_h}{d_{pq}^2} \qquad (4.6)$$

ここに、f_k ：第２最短経路としての代替分を含むリンク k の評価値

x_h ：リンク h がネットワークに含まれないとき $x_h=0$ 、ネットワークに含まれるとき $x_h=1$ とする。

γ_{kh} ：リンク k がリンク h の両端のノード間の第２最短経路に含まれるとき１、含まれないとき０とする。

p, q ：リンク h の両端のノード

第２最短経路だけでなくもっと多くの経路を考慮するか、あるいはネットワークの状態に応じてさらに綿密な評価値をつくることもできる。しかし、大規模問題を解くのに使う場合、複雑な評価値を設けることは計算時間の増加につながるので好ましくない。そして、必要な計算機の記憶容量が大きくなりすぎないようにも注意しなければならない。最短生成木法よりは改良されていても、この評価値は直接に目的関数値の変化を考えたものではない。さらに複雑な評価値にするのならば、むしろ backward 法などのように目的関数値を使った方がよいであろう。

ここで作成しようとしている近似解法は、backward 法などより大規模な問題に適用でき、最短生成木法よりも解の精度が高く応用の可能性も大きいものである。そこで、ここで提案するリンク評価値は、目的関数値が変わったときにも応用できるかを検討する。もっとも簡単で、かつ重要な問題の拡張は需要交通量を扱うことである。この場合には総走行距離の最小化が目的関数になる。これに対応させるには、リンク評価値を求めるとき、最短距離と第２最短距離の差に需要交通量を乗じればよい。このような問題の拡張には容易に応じることができる。すなわち、こ

のリンク評価値を用いれば、最短生成木法よりは応用の可能性のある近似解法になる。

4.4.3　簡易 forward 法の提案

ここでは、4.4.2で提案したリンク評価値を使い、forward 法の手順で解を求める近似解法を提案する。このリンク評価値は、forward 法や backward 法のようにネットワークに含めるリンク、あるいは除くリンクを1本ずつ決める方法を前提としている。リンク評価値を使う場合には、このどちらかの方法を用いることになる。そのうち、ここでは forward 法の手順を用いる近似解法を示す。

forward 法は最短生成木を初期ネットワークとし、リンクを1本ずつ付け加えてゆくものである。計算手順は4.2.3で示したとおりである。リンク評価値を使う場合も同じ手順でよいが、最短生成木を初期ネットワークにする必要はない。式（4.5）、（4.6）でリンク評価値を求めるとき、ネットワークが非連結であってもかまわない。そこで、リンクがまったくない状態を初期ネットワークとし、最初からリンク評価値を使って固定してゆく。解のネットワークは連結でなければならないので、まずリンク評価値の大きいリンクを順に固定して生成木をつくる。そして、その後はリンク評価値の大きいリンクを順に付加してゆく。

この解法は、forward 法の目的関数値を計算する部分を簡単なリンク評価値の計算に置き換えたものといえる。そこで、これを簡易 forward 法と呼ぶことにする。簡易 forward 法のアルゴリズムは次のようになる。

ステップ1：最大ネットワークでノード間の最短経路と第2最短経路およびそれらの長さを求める。そして、それを用いて各リンクの最短経路としての評価値を計算する。

ステップ2：すべてのリンクが含まれない状態を初期ネットワークとする。

ステップ3：ネットワークに含まれず、その両端のノードがたがいにつながれていないリンクのリンク評価値を求める。

ステップ4：リンク評価値が最大のリンクをネットワークに付け加える。生成木ができればステップ5へ進み、そうでなければステップ3へもどる。

ステップ5：ネットワークに含まれず、ネットワークに付け加えてもネットワーク長が制約値を超えないリンクについて、リンク評価値を求める。

ステップ6：リンク評価値が最大のリンクをネットワークに付け加える。ステップ5へもどる。

このアルゴリズムでは生成木ができてステップ5へ進んだとき、すでにネットワーク長が制約値を超えていることが生じ得る。この場合は実行可能解が見つからなかったことになる。これは、backward 法で実行可能解が得られない場合があるのと同じである。しかし、このようなことが起きるのは、ネットワーク長の制約値が最短生成木の長さに近い場合だけである。もし実行可能解が求まらなければ最短生成木を解にすればよいので、とくにこのための対策は設けない。

簡易 forward 法の特徴は次のとおりである。

① リンク評価値を用いて付加リンクを決めるため、経路探索は最大ネットワークで最短経路と第2最短経路を求めるときに行うだけでよい。

② 生成木を作る段階においてもリンク評価値を使える。

解の目的関数値が必要なければ、経路探索はリンク評価値の計算の際に行うだけである。このときには最短経路だけでなく第2最短経路も探さなくてはならないが、それには最短経路探索の結果を利用できる。このように経路探索を繰り返し行わないので、ネットワークの規模が大きくなっても計算時間の増加はゆるやかであると考えられる。そして、生成木を作る段階にもリンク評価値が使えるため、最短生成木よりは目的関数値の小さい生成木を作れるであろう。この点はforward 法よりも優れており、ネットワーク長の制約値が極めて小さい場合には forward 法より良い解が得られる可能性がある。

4.4.4 簡易 backward 法の提案

ここではリンク評価値を使い、backward 法の手順で解を求める近似解法を提案する。

backward 法は最大ネットワークを初期ネットワークとし、リンクを1本ずつ除いてゆくものである。計算手順は4.2.3で示したとおりである。目的関数値を求める部分をリンク評価値の計算に置き換えれば、ここでも同じ手順が使える。ただし、ネットワークが非連結にならないように注意がいる。backward 法では目的関数値の計算を行うときに最短経路を探す。そうすると、この段階でネットワークが連結かどうかがわかる。しかし、4.4.2で提案したリンク評価値を使う場合、式(4.6)でリンク評価値を計算するだけではネットワークの連結性を調べられない。そこで、ネットワークから除く候補になるリンクに対しては、それを除いても連結網であるかどうかを検査する。

この解法を簡易 backward 法と呼ぶことにする。アルゴリズムは次のようになる。

ステップ1：最大ネットワークでノード間の最短経路と第2最短経路およびそれらの長さを求める。そして、それを用いて各リンクの最短経路としての評価値を計算する。

ステップ2：最大ネットワークを初期ネットワークとする。

ステップ3：ネットワークを構成するリンクについて、リンク評価値を求める。

ステップ4：除いても非連結網にならないリンクのなかで、リンク評価値が最小のリンクをネットワークから除く。もし除けば非連結網になるリンクがあれば、そのリンクは解のネットワークに含めることにする。どのリンクを除いても非連結網になるならば、実行可能解は求まらないが計算を終える。

ステップ5：ネットワーク長が制約値を超えているならば、ステップ3へもどる。制約値以下ならば、ステップ6へ進む。

ステップ6：ネットワークに含まれず、ネットワークに付け加えてもネットワーク長が制約値を超えないリンクについて、リンク評価値を求める。このようなリンクがなければ計算を終える。

ステップ7：リンク評価値が最大のリンクをネットワークに付け加える。ステップ6へもどる。

ステップ4での連結性の検査は最短生成木を求める場合と同じようにする。すなわち、ネットワークに含まれるリンクの両端のノードに同じラベルをつけてゆき、このラベルが1種類だけになるかを調べる。

簡易 backward 法の特徴も、経路探索を最大ネットワークで最短経路と第2最短経路を求める

ためにだけ行えばよいことである。そのため、大規模な問題への適用が可能である。これを簡易 forward 法と比べた場合には、連結性の検査が問題になる。最短生成木法に関して述べたのと同じで、全体の計算量が少ないので連結性の検査に要する計算量の差が相対的に大きくなる。簡易 forward 法では生成木ができれば連結性の検査は必要ないのに対し、簡易 backward 法ではすべての段階で行わなくてはならない。そして、連結性の検査自体もリンクを除いたときに非連結にならないかを調べる方が複雑である。

4．4．5　段階的削減法と局所最適化法への応用

ここでは、簡易 forward 法と簡易 backward 法を厳密解法と組み合わせて用いる近似解法を提案する。

簡易 forward 法と簡易 backward 法は、付加リンクや除去リンクの決定方法以外は forward 法、backward 法と同じ構造である。これらを使えば、ネットワークにリンクが1本ずつ付け加えられ、あるいは1本ずつ除かれる。その結果、forward 法や backward 法と同様にリンクを順序づけることになる。もし、これらのリンク評価値を使う解法の近似解の近くに最適解があることが多ければ、厳密解法と組み合わせて解く段階的削減法や局所最適化法へ応用できる。すなわち、backward 法を用いてリンクの順序づけを行っている部分を簡易 forward 法か簡易 backward 法へ置き換えるのである。そうすれば、計算時間を短くできる。

簡易 forward 法や簡易 backward 法の近似解の近傍に最適解があるかどうかを調べるため、4．3．2で行ったと同じように式（4．1）の指標を計算してみる。対象とする問題は forward 法や backward 法について調べたときと同じ95問題である。

計算の結果、指標の平均値はリンクを簡易 forward 法でネットワークにつけ加えられる順に並べたときに6.0、簡易 backward 法で除かれる順に並べたときに6.2であった。backward 法の場合が10.9であったから、リンク評価値を使う解法の方が近似解と最適解との隔たりが小さい。よって、これらの解法を段階的削減法や局所最適化法へ応用すれば、計算時間が短くなるだけでなく、解の精度が向上する可能性がある。

まず、リンクの段階的削減を行う近似解法への応用を考える。段階的削減を行う近似解法は最大ネットワークからリンクを除いてゆく backward 法的な解法である。そこで、簡易 forward 法の方が指標の値はわずかに小さいが、必要のない部分にまでリンクの順序づけをしなくてよい簡易 backward 法を用いることにする。計算手順は backward 法を用いる場合と同じである。

局所最適化を行う近似解法には指標の値の小さい簡易 forward 法を用いる。リンクを順序づける方向が逆であるが、計算手順は backward 法を用いる場合と同じである。

ここで提案したふたつの近似解法は、それぞれ4．3．3と4．3．4で提案した近似解法と同じ特徴を持つ。そして、計算時間はそれらよりも短く、解の精度は高いことが期待できる。

4．4．6　むすび

この節では、大規模な最適交通網構成問題を解くのに適した近似解法を提案した。forward 法や backward 法から経路探索の回数を減らす方法を考え、そのためのリンク評価値を作った。そ

して、このリンク評価値を使用した簡易 forward 法と簡易 backward 法を提案した。さらに、これらの解法を backward 法の代わりに使い、厳密解法を組み合わせた近似解法も示した。

リンク評価値は forward 法や backward法の計算手順を前提とし、目的関数値を毎回計算せずに付加リンクや除去リンクを決めるために作った。これを使った簡易 forward 法と簡易 backward 法では最大ネットワークで最短経路と第２最短経路を求めるために経路探索を行うだけでよい。そのため、計算時間は最短生成木法よりは長いが、forward 法や backward 法よりは大幅に短いと考えられる。またこのリンク評価値は基本問題の場合だけでなく、拡張した問題へも応用が可能である。

厳密解法と組み合わせた近似解法の作成は、この節で目的としていたことではない。しかし、検討の結果、簡易 forward 法や簡易 backward 法を使えば解の精度の向上が期待できた。これは、計算時間と解の精度の両方を同時に改良できる可能性を示しているのである。そこで、簡易 backward 法を使った段階的削減法と簡易 forward 法を使った局所最適化法の２種類の近似解法を提案した。

４．５　計算例による近似解法の比較

４．５．１　はじめに

ここでは、従来の近似解法と提案した近似解法によって例題を解き、解の精度と必要な計算時間を比較する。

従来の近似解法については、４．２で考察した。しかし、これは実際に計算を行ったのではない。そこで、計算例により検討結果を確認する。そのため、ここでは主要な従来の近似解法、４．３で提案した段階的削減を行う近似解法と局所最適化を行う近似解法、４．４で提案した簡易 forward 法と簡易 backward 法および、それらを厳密解法と組み合わせた近似解法を比較の対象とする。

４．５．２では、比較の対象とする近似解法を説明する。

４．５．３では、計算に用いる例題を示す。

４．５．４では、計算結果を示して考察する。

４．５．２　比較の対象とする近似解法

ここでの目的は、従来の近似解法相互、およびそれらと提案した近似解法を比較することである。比較する項目は解の精度と必要な計算時間である。解の精度は近似解の目的関数値がどれだけ最適解の値に近いかである。解の精度を高めようとすれば一般に計算時間は長くなり、これらを両立させることは容易でない。そこで、計算時間と解の精度を総合して近似解法を評価しなければならない。

従来の近似解法の研究には、いくつかの例題についての計算時間と解の精度を示したものがある。しかし、それらを互いに比較することは困難であり、従来の近似解法の優劣は必ずしも明確ではない。そのため、従来の近似解法のなかで、もっとも優れたものと提案した近似解法を比べるというようにはできない。よって、従来の近似解法の主要なものを集め、これらと提案した近

似解法を比較する。

ここでは、次の近似解法を比較する。

1） 最短生成木法
2） Scott の forward 法
3） Scott の backward 法
4） 飯田の近似解法Ⅱ
5） Dionne・Florian の近似解法
6） forward 法
7） backward 法
8） 段階的削減を行う近似解法
9） 局所最適化を行う近似解法
10） 簡易 forward 法
11） 簡易 backward 法
12） 簡易 backward 法を使い、段階的削減を行う近似解法
13） 簡易 forward 法を使い、局所最適化を行う近似解法

これらの近似解法のアルゴリズムをアルゴリズムＡ１～A13と表わす。それぞれのアルゴリズムについて簡単に説明する。

アルゴリズルＡ１は最短生成木法である。最短生成木法では解を求めるのに目的関数値を計算する必要はない。しかし、他のアルゴリズムにあわせ、最大ネットワークと近似解の目的関数値を計算する。よって、このアルゴリズムでは最短経路探索を２回行うことになる。

アルゴリズムＡ２、Ａ３はそれぞれ Scott の forward 法と backward 法である。Scott は完全連結網からリンクを選択しているが、このアルゴリズムでは与えられた最大ネットワークから選ぶ。そして、リンクも両端のノード番号で識別するのでなく、定められたリンク番号を用いる。

アルゴリズムＡ４は、DP的探索法を用いる飯田の近似解法Ⅱである。

アルゴリズムＡ５は、Dionne らの考え方を用いる近似解法である。Dionne らは Hoang の分枝後退法のアルゴリズムを基礎にしていた。だが、ここでは厳密解法のなかで計算時間が最短であったアルゴリズムＥ３をもとにして修正を加える。アルゴリズルＥ３は連結網化分枝を使用するもので、段階的削減や局所最適化を行う近似解法にも組み込んでいる。アルゴリズムＡ５は目的関数値の下限値を求める部分を除き、アルゴリズムＥ３とまったく同じである。

アルゴリズムＡ６、Ａ７は、それぞれ Scottの forward 法と backward 法からネットワーク変換操作を取り去ったものである。ただし、アルゴリズムＡ７には４．３．３で述べた改良を加えている。これらは従来の実用的な近似解法のなかで、もっとも簡単なものである。そして、これらとアルゴリズムＡ２、Ａ３の計算結果を比べれば、これまで明確に示されていなかったネットワーク変換操作の効果を調べることができる。

アルゴリズムＡ８～A13は、４．３と４．４で提案した近似解法である。そのアルゴリズムについてはすでに述べたとおりである。

従来の近似解法は発表されているアルゴリズムをもとに作成したものである。ただし、ネットワークが非連結であることがわかれば最短経路探索を省略するなど、基本的な考え方を損わない範囲で改良を加えている。計算機を用いる場合のネットワークの表現方法や最短経路探索のアルゴリズムなどは、すべて同じものを使う。これは、最適交通網構成問題の近似解法としての基本的な部分以外での差ができるだけ生じないようにするためである。

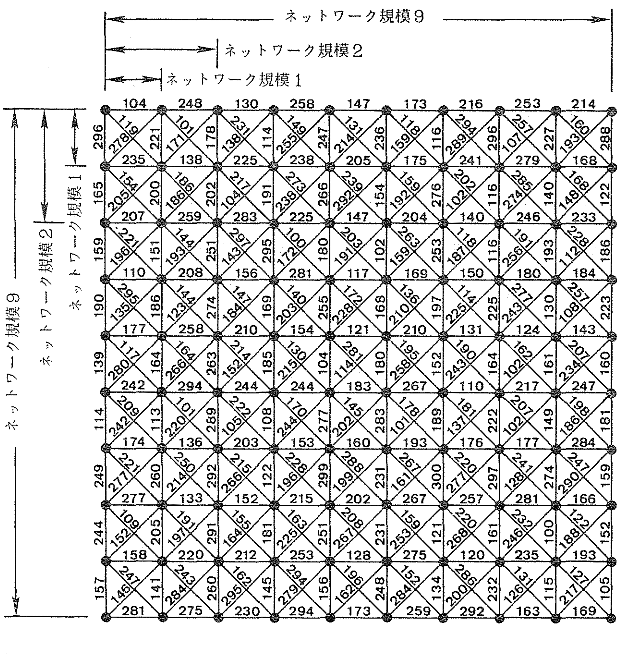
比較対象の近似解法は13種類であるが、参考のためにこれに厳密解法を加える。連結網化分枝を行うアルゴリズムE３でも同じ計算を行い、これらの近似解法での計算時間と比較する。

４．５．３　計算に用いる例題

近似解法の評価は解の精度と計算時間を総合して行わなければならない。そして、計算時間については大規模な問題へも適用できるかどうかが重要である。そこで、まず計算時間と解の精度をみるために、第３章で厳密解法のアルゴリズムの比較に用いた５種類の例題を与える。つぎに、ネットワークの規模が大きくなったときの計算時間の変化を調べるための例題を与える。

例題３．１～３．５に対してネットワーク長の制約水準を10%、25%、40%、55%、70%、85%とし、それぞれ６問題を解く。最大ネットワークやリンク長、ネットワーク長の制約値は**３．５．３**で示したとおりである。

ネットワークの規模が大きくなったときの計算時間の変化をみるためには、図－４．５の例題



図―4.5　例題4.1の最大ネットワーク

４．１のネットワークを使う。このネットワークの一部分を使い、同じネットワークパターンでネットワークを構成するノード数を変えるのである。リンクの長さは**図－４．５**に示したとおりである。これは、それぞれ $100 \leqq l_k \leqq 300$ で、各リンクはその両端のノード間の最短経路になるように乱数を使って決めたものである。図のようにネットワーク規模を１、２、．．．．、９と変えてそれぞれを最大ネットワークとし、制約水準を50％とした問題を与える。各問題のノード数、リンク数などは**表－４．７**に示す。

表―4.7　例題4.1のノード数，リンク数など

ネットワーク規模	ノード数	リンク数	最大ネットワーク長	ネットワーク長の制約値
1	4	6	1253	855
2	9	20	3853	2519
3	16	42	8010	5011
4	25	72	14136	8752
5	36	110	21378	13046
6	49	156	29950	18147
7	64	210	41367	25032
8	81	272	53825	32285
9	100	342	67834	40581

４．５．４　計算結果と考察

ここでは例題の計算結果を示し、その考察を行う。

例題を解くにあたり、アルゴリズムＥ３、Ａ５、Ａ８、Ａ９、A12、A13の厳密解法の部分では、目的関数の下限値を強める操作をリンクを３本除くごとに行うことにする。これは第３章での計算例と同じである。また、アルゴリズムＡ８、Ａ９、A12、A13では、それぞれネットワーク長の制約値の減少幅と検討リンクの範囲のパラメータを決めなくてはならない。これらのパラメータ値による変化を調べられるように、制約値の減少幅は平均リンク長の１、２、．．．、６倍とし、検討リンクの範囲は１、２、．．．、６とする。すなわち、これらのアルゴリズムではパラメータの値を変え、６回ずつ問題を解く。

例題の計算には神戸大学総合情報処理センターのACOS900を使用した。例題３．１～３．５を解くのに要した計算時間は**表－４．８**に示すとおりである。**表－４．８**では、各例題ごとにネットワーク長の制約水準の異なる６問題を解いた合計の計算時間を示してある。そして、最短生成木法であるアルゴリズムＡ１と比べてどれだけの計算時間がかかるかを比

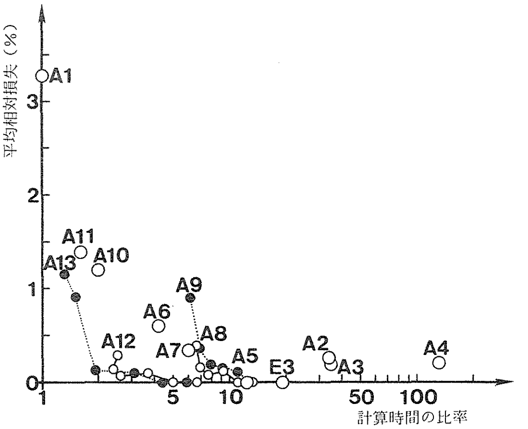
表—4.8　例題3.1～3.5を解くのに要した計算時間

アルゴリズム	計算時間(ms) 例題3.1	例題3.2	例題3.3	例題3.4	例題3.5	合計(s)	計算時間の比率
E3	666	2015	3683	16091	16981	39.4	18.91
A1	91	104	122	141	151	0.6	1
A2	1683	2815	6924	9337	9535	30.3	33.81
A3	1773	3431	6576	7069	10409	29.3	34.36
A4	6061	12378	21497	26748	39429	106.1	129.74
A5	546	1420	2104	10523	11911	26.5	12.28
A6	295	430	647	722	930	3.0	4.17
A7	340	607	934	1026	1244	4.2	6.05
A8—1	367	663	1006	1164	1338	4.5	6.67
2	371	678	1073	1237	1362	4.7	6.92
3	400	761	1177	1388	1548	5.3	7.68
4	467	953	1340	1928	1899	6.6	9.24
5	533	1088	1624	2352	2254	7.9	10.86
6	605	1396	2297	3173	2770	10.2	13.19
A9—1	322	586	960	1068	1258	4.2	6.17
2	359	658	1042	1202	1366	4.6	6.91
3	400	744	1181	1367	1516	5.2	7.82
4	458	873	1355	1670	1739	6.1	9.10
5	556	1051	1585	2091	2114	7.4	10.91
6	609	1297	1952	2663	2560	9.1	12.88
A10	173	213	268	259	290	1.2	1.99
A11	135	175	206	230	237	1.0	1.61
A12—1	183	269	350	427	417	1.6	2.54
2	174	259	326	408	398	1.6	2.37
3	172	299	370	535	451	1.8	2.61
4	232	446	512	1025	674	2.9	3.67
5	279	606	783	1517	1088	4.3	4.94
6	358	972	1610	2183	1668	6.8	6.71
A13—1	103	144	172	182	194	0.8	1.31
2	116	166	195	221	213	0.9	1.50
3	122	219	252	358	276	1.2	1.92
4	167	362	489	613	465	2.1	3.07
5	234	555	734	1061	709	3.3.	4.39
6	299	770	983	1732	1311	5.1	5.88

率で表わした。この比率は各問題ごとの値を相乗平均したものである。解の精度に関しては、得られた近似解が最適解でなかった問題の数、最大相対損失、それに平均相対損失を表－４．９に示す。計算時間と解の精度との関係を表わしたのが図－４．６である。この図は、計算時

表―4.9　例題3.1～3.5の近似解の精度

アルゴリズム	最適解でない近似解の数	最大相対損失（%）	平均相対損失（%）
A 1	26	18.96	3.27
A 2	10	3.55	0.26
A 3	9	3.55	0.19
A 4	10	3.00	0.21
A 5	0	0	0
A 6	13	9.11	0.61
A 7	11	5.52	0.34
A 8―1	15	3.00	0.39
2	7	3.00	0.16
3	4	1.01	0.08
4	2	3.00	0.12
5	0	0	0
6	0	0	0
A 9―1	18	6.72	0.90
2	14	3.00	0.37
3	7	3.00	0.19
4	4	3.00	0.15
5	3	2.54	0.11
6	0	0	0
A10	13	23.06	1.20
A11	14	25.50	1.39
A12―1	10	5.13	0.29
2	4	3.00	0.14
3	3	1.01	0.07
4	1	3.00	0.10
5	0	0	0
6	0	0	0
A13―1	11	25.37	1.15
2	6	21.14	0.91
3	3	3.00	0.13
4	1	3.00	0.10
5	0	0	0
6	0	0	0



図―4.6　アルゴリズムＡ１に対する計算時間の比率と平均相対損失との関係

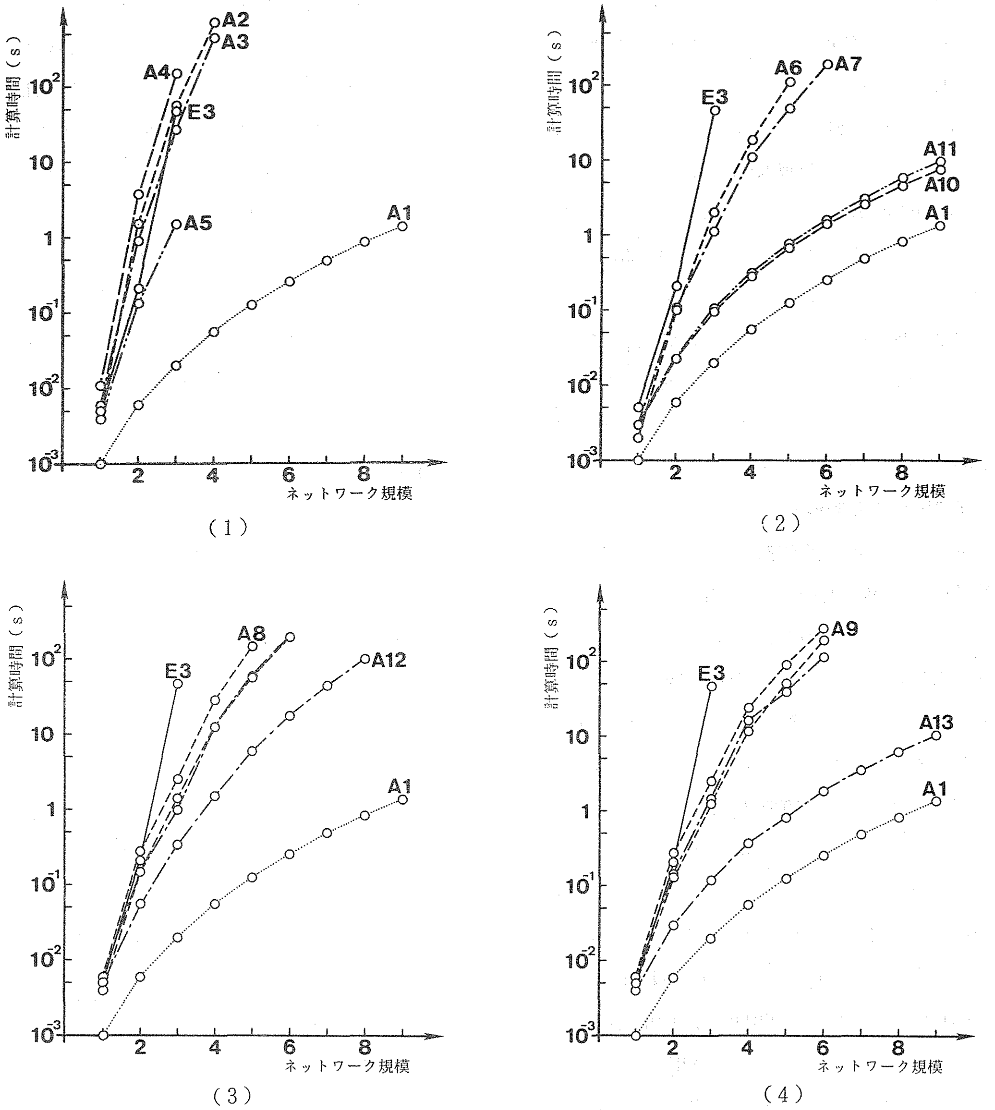
間の比率と平均相対損失との関係を示し、左下方にあるものが計算時間が短く、解の精度が高い。

例題4．1の計算結果は**表－4．10**のようになった。この例題の計算の際は、計算時間が100秒を超えれば、そのアルゴリズムではそれよりも規模の大きいネットワークの問題を解いていない。またそうでないときも、1000秒を超えるときには計算を打ち切った。ネットワーク規模によ

表―4.10　例題4.1を解くのに要した計算時間　　(単位：ms)

アルゴリズム	計算時間 規模1	規模2	規模3	規模4	規模5	規模6	規模7	規模8	規模9
			(×10)	(×10)	(×10)	(×10)	(×10)	(×10)	(×10)
E 3	5	209	4756						
A 1	1	6	2	6	13	26	49	83	137
A 2	4	1478	5568	71780					
A 3	6	889	2737	44627					
A 4	11	3803	15115						
A 5	4	132	146						
A 6	2	101	206	1909	11284				
A 7	3	107	114	1112	5025	19203			
A 8－1	6	146	140	1249	5682	20265			
2	5	165	151	1326	6087	21395			
3	6	195	182	1455	6461	22191			
4	6	211	211	1756	7503	27250			
5	6	259	218	2535	8985	32317			
6	4	275	253	2865	15069				
A 9－1	5	130	128	1177	5190	19521			
2	6	159	142	1231	5451	20499			
3	6	194	158	1304	5614	20503			
4	5	224	178	1442	5943	21305			
5	5	260	208	1659	6685	22984			
6	5	275	251	2419	9250	28076			
A10	3	23	10	28	69	144	264	466	766
A11	3	24	10	31	77	162	319	606	983
A12－1	5	55	39	173	602	1786	4476	10459	
2	5	58	34	153	611	1788	4459	10168	
3	6	67	43	228	791	2457	7036	17491	
4	5	68	53	373	1298	5212	13362		
5	5	150	87	906	2836	9913	22496		
6	6	149	100	1270	5961	19694			
A13－1	4	30	12	37	82	186	352	623	1048
2	5	34	14	43	95	235	370	777	1299
3	5	51	23	71	159	421	626	1397	2294
4	5	119	49	128	421	937	1682	3218	5370
5	6	143	84	453	1285	3457	4638	9069	13128
6	5	140	149	1620	3938	11706			

る計算時間の変化を表わしたのが図－４．７である。ただし、アルゴリズムＡ８、Ａ９、A12、



図— 4.7　ネットワーク規模による計算時間の変化

A13では、パラメータ値の異なる６種類の計算結果の最短計算時間と最長計算時間だけを示している。

これらの図表をもとに、各アルゴリズムでの計算結果を考察する。なおアルゴリズムＡ１～A13をＡ１～A13だけで表わす。

最短生成木法であるＡ１による計算時間はもっとも短く、ネットワークの規模が大きくなれば、他のものとの差は急激に拡大する。平均相対損失は3.3%で、解の精度は他の近似解法よりもかなり悪い。

Scott の forward 法と backward 法であるＡ２とＡ３による計算結果はよく似ている。計算時間は厳密解法であるＥ３と差がないといえる。ネットワークの規模が大きくなったときの計算時間の増加も急激であり、大規模な問題への適用は困難であることがわかる。一方、解の精度はＡ１に比べればはるかによい。これらからネットワーク変換操作を除いたＡ６、Ａ７の計算時間は、従来の近似解法のなかではＡ１についで短い。そのかわり解の精度は悪い。４種類の計算結果からネットワーク変換操作の効果をみると、相対損失が約½に減っている。そのために計算時間は７～10倍を要している。例題４．１の計算結果からは、ネットワーク規模が大きくなれば、ネットワーク変換操作に必要な計算時間が急激に増加することがわかる。これらを考えれば、この操作は有効な方法でないといえる。

飯田の近似解法IIであるＡ４は必要な計算時間がもっとも長く、厳密解法を上回っている。ネットワークの規模が大きくなったときの計算時間の増加も急激である。この計算結果からみるかぎり、近似解法として用いる利点はない。解の精度はＡ２、Ａ３と同程度である。

Dionne らの近似解法であるＡ５ではすべての問題で最適解が得られた。解の精度は非常に高い。計算時間は厳密解法の70％程度である。Ａ１に対する計算時間の比率をみると、Ａ２、Ａ３よりも低く、従来の近似解法のなかでは優れている。しかし、ネットワークの規模が大きくなれば計算時間は急激に増加する。図－４．７では計算時間の変化はゆるやかに思えるが、ネットワーク規模が４のときに1000秒を超えている。したがって、大規模な問題を解くことはできない。

段階的削減を行うＡ８と局所最適化を行うＡ９の計算結果は類似している。Ａ９ではパラメータの値が大きいほど計算時間は長く、解の精度は高くなっている。Ａ８ではパラメータ値が大きい方が解の精度が悪いこともあるが、全体の傾向はＡ９と同じである。パラメータの値が小さいときには解の精度と計算時間はＡ７と差がなく、大きいときにはＡ５とほぼ同じになっている。そして、パラメータ値を一定にすれば、ネットワークの規模が大きくなっても計算時間の増加はゆるやかである。これらのことから、Ａ８、Ａ９は従来の近似解法より優れているといえる。

大規模問題を解くためのA10、A11は、その開発の目標どおりにＡ１とＡ６、Ａ７の中間の性格を持つことが示されている。解の精度は平均相対損失が1.2％、1.4％と悪いが、これは例題３．５の制約水準が10％の問題の近似解が最適解からかけ離れているためである。これ以外はＡ６、Ａ７と大差ない。ネットワークの規模が大きくなったときの計算時間の増加もゆるやかである。例題４．１の計算結果でA11の計算時間と最大ネットワークのノード数 n との関係を調べると、n^2 よりは大きく変化し、n^3 よりはゆるやかである。Ａ７について調べると、４．２．６で予想したとおり n^4 と n^5 の間である。このことから、A10、A11はＡ６、Ａ７と比べるとはるかに大規模な問題へ適用できることが実証された。

リンク評価値を用いる解法を使い、段階的削減あるいは局所最適化を行うA12、A13は計算時間が短く、しかも解の精度も高くできるという結果が得られている。これらはパラメータ値を変えれば、Ａ８、Ａ９と同様の変化を示す。図－４．７ではA12、A13は他のアルゴリズムの左下方に位置し、計算時間と解の精度の両面で優れていることがわかる。また、A13はパラメータ値が小さいとき、基礎になっているA10よりも計算時間が短い。これは、A13の簡易 forward 法の部分では、リンクを付け加えてネットワーク長の制約値を超えないかを、それぞれのリンクに

ついて調べないからである。ここでの計算結果からは、A12、A13は実用的な近似解法のなかで、もっとも優れているといえる。

例題の計算結果をもとにした近似解法の比較、考察をまとめると、次のようになる。

① 最短生成木法は計算時間が短いが、解の精度は悪い。

② Scott の近似解法はネットワーク変換操作に要する計算時間が長く、大規模な問題を解くことは困難である。

③ 飯田の近似解法Ⅱの解の精度は Scott の近似解法と同程度である。しかし、計算時間は厳密解法よりも長い。

④ Dionne らの考え方に基づく近似解法は解の精度が非常に高い。しかし、大規模な問題には適用できない。

⑤ 段階的削減を行う近似解法と局所最適化を行う近似解法は、従来の近似解法と比べて解の精度、計算時間ともに改善されている。とくにリンク評価値を用いる簡易 forward 法、簡易 backward 法と組み合わせれば、極めて短い計算時間で精度の高い解を求めることができる。そして、大規模な問題への適用も可能になる。

⑥ リンク評価値を用いる簡易 forward 法と簡易 backward 法は、最短生成木法と forward 法、backward 法の中間の性格を持つ。forward 法や backward 法と比べれば、はるかに大規模な問題へ適用できる。

4.5.5 むすび

この節では、従来の近似解法と提案した近似解法によって例題を解き、解の精度と計算時間を比較した。比較の対象としたのは合計13種類の近似解法である。まず解の精度と計算時間の関係をみるために、5種類の例題の30問題を解いた。つぎに、ネットワーク規模が大きくなったときの計算時間の変化を調べるための例題を解いた。そして、これらの計算結果を考察した。

従来の近似解法の計算結果は、4.2での検討結果を裏づけるものであった。計算効率が悪いと予想された Scott のネットワーク変換操作や飯田の近似解法Ⅱは計算時間が長かった。ここでの計算結果では、改良された厳密解法の方が近似解法の計算効率が悪いものより計算時間が短い場合もあった。従来の近似解法で優れているのは Dionne らの近似解法であった。

提案した近似解法は、いずれも開発の目標を達成する結果を示した。すなわち、リンクの段階的削減を行う近似解法と局所最適化を行う近似解法は、従来よりも短い計算時間で精度の高い近似解を求めることができた。簡易 forward 法と簡易 backward 法は大規模な問題へ適用できることが示された。そして、これらを組み合わせた近似解法では、極めて短い計算時間で精度の高い解が得られた。

4.6 結　語

この章での目的は、短い計算時間で最適解ないしはそれに近い解を求められる近似解法を作成することであった。そのために従来の近似解法を検討し、いくつかの近似解法を提案した。

従来の近似解法の検討結果として、次のことを得た。

① 最短生成木法はもっとも簡単な近似解法であるが、拡張した問題への適用は困難であり、実用的な価値はほとんどない。

② Scott の近似解法は、その特徴であるネットワーク変換操作の計算効率が悪く、大規模な問題を解くのに適さない。しかし、forward 法と backward 法の考え方は、拡張した問題へも適用できる。

③ 飯田の2種類の近似解法は、単純な backward 法よりは解の精度が高い。しかし、むだな計算を繰り返す可能性があり、必要な計算時間は長い。

④ Dionne らの近似解法は解の精度が高く、むだな計算を繰り返すこともないと考えられる。しかし、厳密解法の欠点を引き継ぎ、大規模な問題への適用は困難であると予想できる。

⑤ 近似解法を作成する際には厳密解法での計算例の考察を生かし、厳密解法の応用を図るのがよい。また、backward 法などよりも大規模な問題を解ける近似解法を作成する必要がある。

これらの検討結果をもとに、厳密解法を応用したものとして、段階的削減を行う近似解法と局所最適化を行う近似解法を提案した。そして、大規模問題を解くためのものとして、最大ネットワークでの経路探索で求めた値からリンク評価値を計算し、forward 法あるいは backward 法の手順を用いる近似解法を提案した。さらに、このリンク評価値を使用する解法を厳密解法を応用した解法に組み合わせたものも提案した。

提案したものに従来の近似解法をあわせ、13種類の近似解法で例題を解き、解の精度と計算時間の比較を行った。その結果、従来の近似解法については前述の検討結果が裏づけられた。そして、提案した近似解法による計算結果からは、次のことが明らかになった。

① 段階的削減を行う近似解法と局所最適化を行う近似解法は、従来の近似解法よりも短い計算時間で精度の高い解を求められる。そして、リンク評価値を用いる近似解法と組み合わせれば、計算時間はさらに短くなり、大規模な問題へも適用できる。

② リンク評価値を用いる簡易 forward 法と簡易 backward 法は、計算時間と解の精度に関し、最短生成木法と forward 法、backward 法との中間の性格を持つ。backward 法などよりは大規模な問題へ適用できる。

ここで提案した近似解法は、それぞれの開発の目的を果たしたといえる。段階的削減を行う近似解法と局所最適化を行う近似解法は、パラメータの値を変えることにより必要な計算時間と解の精度を制御できる。これは、実用上においては有用な特徴である。これにリンク評価値を用いる解法を組み合わせたものは、従来のすべての近似解法よりも優れているという計算結果であった。よって、短い計算時間で精度の高い解を求められる近似解法を作成するという目的は達せられた。

参考文献

1) Dionne,R. and M.Florian : Exact and approximate algorithms for optimal network design, Networks, Vol. 9, №1, pp.37〜59, 1979.

2) Scott,A.J. : The optimal network problem : some computational procedures, Transportation Research, Vol. 3, pp.201〜210, 1969.

3) 飯田恭敬：最適道路ネットワークの構成手法、土木学会論文報告集、第241号、pp.135〜144, 1975年9月.

4) 日本オペレーションズ・リサーチ学会：ネットワーク構造を有するオペレーションズ・リサーチ問題の電算機処理に関する基礎研究、報文シリーズ・T−73−1, 1973年3月.

5) 枝村俊郎・森津秀夫：最適交通ネットワーク問題の厳密解法と近似解法、土木学会論文報告集、第262号、pp.113〜127, 1977年6月.

6) 森津秀夫：大規模最適ネットワーク問題への適用を考慮した近似解法、土木学会第35回年次学術講演会講演概要集、第4部、pp.49〜50, 昭和55年9月.

第５章　最適交通網構成問題の基本的な拡張に関する考察

５．１　概　　説

最適交通網構成問題の基本問題は、ネットワークの総延長の制約下で最短距離の合計を最小化するものである。第３章と第４章では、この基本問題を対象に解法を考えてきた。しかし、実際に交通網計画を行うときには、基本問題をそのまま適用できることはほとんどない。これを変形、拡張した問題を扱わなければならないのである。そこで、この章では最適交通網構成問題を拡張する場合の基本的なものを取り上げ、どのような解法の修正が必要かを考察する。

基本問題の拡張に関しては、すでに２．５で述べた。そのうち、ここでは複数の制約条件を持つ問題、多目的最適交通網構成問題、交通網の段階建設問題、それに不確実な需要交通量を用いる問題を取り上げる。

５．２では、複数の制約条件を持つ最適交通網構成問題の解法を検討する。そして、多次元ナップザック問題と同じ制約条件を持つ場合を対象とする解法を提案する。

５．３では、複数の目的関数を持つ多目的最適交通網構成問題を対象とし、目的関数のスカラ化手法を用いる解法を提案する。

５．４では、交通網を段階的に建設する問題を最適交通網構成問題として定式化し、解法を提案する。

５．５では、不確実な需要交通量を用いて最適交通網を決定する問題を示し、その解法を提案する。

５．６では、この章で得られた成果をまとめる。

５．２　複数制約条件の問題への拡張

５．２．１　はじめに

最適交通網構成問題の基本問題の制約条件は、ネットワークの総延長に関するものだけである。これに似たものには、ネットワークの建設費だけを制約条件とする問題がある。しかし、現実の交通網計画問題では、ただひとつの制約条件だけでなく、多くの複雑な制約条件を考えなければならないこともある。このような場合、基本問題を対象に作られた解法をそのまま用いることはできない。そこで、この節では複数の制約条件を持つ最適交通網構成問題を解くために、基本問題に対する解法をどのように直さなければならないかを考察する。

まず複数の制約条件を持つ最適交通網構成問題の例を示し、その解法を検討する。つぎに、複数の制約条件の問題に多い、多次元ナップザック問題と同じ制約条件を持つ場合を対象とする解法を提案する。そして、その具体的な問題を例題とし、計算例を示す。

５．２．２では、どのようなときに複数の制約条件を持つ最適交通網構成問題を考えなければならないかを述べる。そして問題を分類し、それぞれの場合の解法を検討する。

５．２．３では、制約式が多次元ナップザック問題と同じ形である最適交通網構成問題を取り上げ、多次元ナップザック問題の近似解法を応用した解法を提案する。

５．２．４では、５．２．３で取り上げた問題の具体的な例として列車の運行パターンを最適

化する問題を示し、提案した解法を適用する。

5.2.2　複数の制約条件を持つ最適交通網構成問題

ここでは、最適交通網構成問題において複数の制約条件を考慮しなければならないのはどのような場合か述べ、その解法について基礎的な考察を行う。

最適交通網構成問題の基本問題は、ネットワークの総延長の制約下で最短距離の合計を最小化するものであった。これに近い実際的な問題には、建設費の予算制約下で総走行距離を最小化するものがある。この問題には基本問題の解法をほとんどそのまま適用できる。しかし、現実には単に予算制約だけでなく、他の制約条件を同時に考えなければならないことが多い。たとえば道路網計画の場合には、次のような制約条件が考えられる。

①　特定のノード間の最短距離の制約

②　リンクの交通量に対する容量制約

③　リンク沿道あるいはネットワーク全体の騒音や排出ガスの制約

④　特定のリンクの組み合わせに関する制約

①〜③は最短距離、リンク交通量、騒音や排出ガスというようなネットワークによって定まる関数値に対する制約条件である。④は決定変数間の関係を規定するもので、直接にネットワークの形状を決める。具体的には3本のリンクのうちの1本だけを必ずつくるとか、リンク k をつくらないならばリンク h もつくらないというようなものである。予算制約もこの種の制約条件であるといえる。

この例のように、交通網計画で使われる制約条件にはさまざまなものがある。これらはネットワークの関数値に対する制約とリンクの組み合わせに関する制約に大別できる。そして、これらの制約条件の組み合わせにより、いろいろな最適交通網構成問題ができるのである。

どのような制約条件を持つ問題でも、リンクの組み合わせを調べて解を求めることは同じである。厳密解法を用いても、あるいは近似解法を用いても、制約条件を満たすリンクの組み合わせである実行可能解からひとつの解を選択するのである。したがって、まず実行可能解を求めることが必要である。そのためには、どのリンクを組み合わせれば実行可能解になるのかがわからなくてはならない。解を得ることの容易さは、実行可能解の集合が単純な構成であるかどうかで決まる。

実行可能解を探す場合、決定変数の値を変えると個々の制約式の値がどのように変化するかを調べる。このとき、リンクの組み合わせに関する制約式では、決定変数の値を代入して直接に制約式の値を計算できる。しかし、ネットワークの関数値に対する制約式では、経路探索や交通配分を行わなければ制約式の値を計算できない。この点において、ネットワークの関数値に対する制約式は取り扱いにくい。

ネットワークの関数値でも、ノード間の距離のように、リンクの除去によって減少しないことが明らかなものがある。この場合は、制約式を満たすためにリンクを除けばよいのか、加えればよいのかだけはわかる。しかし、騒音の影響の大きさのように、リンクの除去によって値が増加するのか減少するのかわからないものもある。このように、リンクの除去と制約式の値の増加、減少の関係が明確かどうかにより、ネットワークの関数値に対する制約式でも取り扱いの容易さ

が異なる。

実行可能解を探す際の取り扱いの容易さから制約式を分類すると、次のようになる。

① リンクの組み合わせに関する制約式

② リンクの除去あるいは付加と、その値の増加、減少の関係が明確なネットワークの関数値に対する制約式

③ リンクの除去あるいは付加と、その値の増加、減少の関係が明確でないネットワークの関数値に対する制約式

つぎに、実行可能解の探索の容易さを制約条件全体で考える。基本問題はネットワーク長の制約式だけなので、リンクを除けば実行可能な方向へ進み、どれだけ実行可能解に近づくかがはっきりしていた。これと同じ状態であれば、制約条件が複数であっても簡単に実行可能解を探せる。つまり、リンクを除くことにより制約式の値がすべて実行可能な方向へ変化する問題であれば、実行可能解の探索が容易である。そして、基本問題と類似の解法が適用できる。また、リンクの組み合わせに関する制約式ばかりであれば、リンクの除去が必ずしも実行可能解に近づくことにならなくても、実行可能解の探索は困難でない。

これらのことから制約条件の組み合わせを次に示す4種類に分ける。そして、それぞれの場合に解法をどのようにすればよいかを考える。

① リンクの組み合わせに関する制約式だけからなり、リンクを除けばすべての制約式の値が実行可能な方向へ変化する場合。

② ネットワークの関数値に対する制約式を含むが、リンクを除けばすべての制約式の値が実行可能な方向へ変化する場合。

③ リンクの組み合わせに関する制約式だけからなるが、リンクを除けば制約式の値が実行不可能な方向へ変化することのあるものを含む場合。

④ リンクの除去と制約式の値の変化との関係が明確でないネットワークの関数値に対する制約式を含む場合。あるいは、リンクを除けば制約式の値が実行不可能な方向へ変化することのある、リンクの組み合わせに関する制約式と、ネットワークの関数値に対する制約式を含む場合。

①の場合には基本問題に対するのと同じ考え方で解法を作成できる。厳密解法では、目的関数の下限値を求めるための補助問題や分枝変数の選択方法は修正が必要である。しかし、すべての制約式が満たされるまでリンクを除くように分枝し、つぎにあともどりすることに変わりはない。近似解法においても、forward 法や backward 法などの基礎的な解法を使える。したがって、この場合の問題には解を求めるのに大きな困難はない。

②の場合の問題も、①の場合と同じような分枝方法で最適解を求められる。また近似解法でも forward 法や backward 法を使える。だが、計算途中の段階で解が実行可能かどうかを調べるときには経路探索や交通配分を行わなければならない。そして、どこまで分枝すれば実行可能になるかをあらかじめ知ることは困難である。そのため、探索木の節点から分枝して得られる実行可能解の目的関数の下限値を求めることはできないであろう。このように、この場合の問題には基本問題の解法に近いものが使えるが、解を求めるのに必要な計算量は大きいと予想できる。な

お、ネットワークの関数値に対する制約式にはネットワークの機能を確保するためのものが多いので、ここに示した場合の問題は少ないと考えられる。

③の場合の問題では実行可能解の集合は単純な構成でない。リンクを除いてゆけば必ず実行可能解に達するのではない。そのため、基本問題の解法とは解の探索方法を変えなければならない。しかし、どのようなリンクを組み合わせれば実行可能になるかは容易にわかるので、解を求めることは必ずしも困難ではない。この場合の制約条件は、一般の０－１整数計画問題とまったく同じである。そこで、それらの問題に対する解法を応用できる。目的関数値の計算の部分を直し、最適交通網構成問題の特性を利用するように修正すればよい。近似解法も forward 法や backward 法はそのままでは使えないので、新しいものを作成しなければならない。

④の場合の問題では実行可能解の集合を捕えることが容易でない。最適解を求めるには完全列挙法に近い解法を使用しなくてはならない。近似解法も計算時間のかかるものか、最適解には遠い近似解しか得られないものになることが予想できる。この場合の問題では、それぞれの問題の特性をいかに利用するかが、解法を作成する際にとくに重要である。

以上のように、ここでは複数の制約条件を持つ最適交通網構成問題の解法の基礎的な考察を行った。制約式を３種類に分け、その組み合わせと基本問題の解法との関係によって４種類の場合の問題を考えた。考察の結果をまとめると、次のようになる。

① 基本問題と類似の解法が使えるのは、リンクを除けばすべての制約式の値が実行可能な方向へ変化する場合である。その場合でも、ネットワークの関数値に対する制約式を含むときは、計算時間を短縮する分枝方法の適用が困難である。

② リンクの組み合わせに関する制約式だけからなるが、リンクを除けば制約式の値が実行不可能な方向へ変化することのあるものを含む場合は、一般の０－１整数計画問題の解法の応用が考えられる。

③ これら以外の場合には、完全列挙法に近い解法を用いなくてはならない。それぞれの問題の特性を利用することが、とくに重要である。

５．２．３　多次元ナップザック問題の近似解法の応用[1)]

ここでは、制約式が多次元ナップザック問題と同じ形である最適交通網構成問題の解法を提案する。

複数の制約条件を持つ最適交通網構成問題のなかで、比較的容易に解を求められるのは制約式がリンクの組み合わせに関するものだけで、リンクを除けば必ず実行可能解に近づくものであった。そこで、ここではこの問題を取り上げ、その解法を示す。目的関数を総走行距離の最小化とすれば、対象とする問題は次のように定式化できる。

問題５．１

$$\min \quad Z=\sum_i \sum_j q_{ij}\, d_{ij}(\boldsymbol{x}) \qquad (5.1)$$

$$\text{s.t.} \quad \sum_k a_{kl}\, x_k \leqq b_l \qquad (5.2)$$

$$x_k = 0 \ \mathit{or}\ 1 \qquad (5.3)$$

ここに　Z：総走行距離

q_{ij}：ノード ij 間のトリップ数

d_{ij}：ノード ij 間の最短距離

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_k \cdots\}$

：ネットワークの状態を表わし、リンク k がネットワークに含まれるとき $x_k=1$ 、含まれないとき $x_k=0$ とする。

a_{kl}：制約式 l における変数 k の係数を表わし、 $a_{kl} \geqq 0$ 、すなわち非負とする。

b_l：制約式 l の上限値を表わす定数で、 $b_l \geqq 0$ とする。

式（５．１）は総走行距離の最小化を表わす目的関数である。式（５．２）の制約式がひとつであれば基本問題と類似の問題であるが、ここでは複数あるものとする。

問題５．１の制約式は多次元ナップザック問題の制約式と同じ形をしている。これは、最適交通網構成問題の基本問題の制約式がナップザック問題の制約式と同じであることに対応している。このことは、基本問題において、目的関数の下限値を求めるための補助問題にナップザック問題を使ったように、問題５．１では多次元ナップザック問題を利用できる可能性があることを示している。

まず厳密解法を作成する。基本問題と同じように考えれば、目的関数の下限値を求めるための補助問題は次のようになる。

補助問題５．１．１

$$\min \quad F_1=\sum_k f_k y_k \tag{5.4}$$

$$\text{s.t.} \quad \sum_k a_{kl}\, y_k \geqq \sum_k a_{kl} - b_l \tag{5.5}$$

$$y_k = 0 \ or \ 1 \tag{5.6}$$

ここに、F_1：実行可能にすることによる目的関数 Z の増加の下限値

f_k：リンク k をネットワークから除くことにより生じる目的関数 Z の増加の下限値

y_k：リンク k がネットワークに含まれるとき $y_k=0$ 、含まれないとき $y_k=1$ とする。

この問題は多次元ナップザック問題である。これを解けば、探索木上の節点から分枝して得られる実行可能解の目的関数値と、最大ネットワークの目的関数値との差の下限値が求まる。基本問題では計算時間を短くするため、ナップザック問題を直接に解かないで、線形計画問題に直して解いた。しかも、その解の性質からシンプレックス法を使わずに簡単に解を得ることができた。しかし、多次元ナップザック問題に対してはそのような解法が示されていない。補助問題５．１．１の整数制約をゆるめた線形計画問題は、シンプレックス法を用いて解かなくてはならない。そして、得られる解も１変数を除いて、他の変数は０か１の値をとるという分枝変数の決定に都合のよいものではない。よって、線形計画問題に直してシンプレックス法で解くのは有効な方法ではない。

基本問題に対しては、実行可能解まで一度に分枝する実行可能化分枝を提案した。ここでも、補助問題５．１．１の最適解ないしはそれに近い近似解が簡単に求まるならば、実行可能化分枝

を採用できる。そこで、実行可能化分枝を行うために、多次元ナップザック問題の近似解法を用いて補助問題を解くことにする。得られる解は近似解なので、その目的関数値を使って問題５．１の目的関数の下限値を定義することはできない。けれども、最適解が早い段階で求まる可能性の高い分枝方法を構成できる。

補助問題５．１．１は通常の多次元ナップザック問題の表現とは異なる。多次元ナップザック問題の近似解法が適用しやすいように定式化し直せば、次の補助問題５．１．２になる。

補助問題５．１．２

$$\max \quad F_2 = \sum_k f_k x_k \qquad (5.7)$$

$$\text{s.t.} \quad \sum_k a_{kl} x_k \leqq b_l \qquad (5.8)$$

$$x_k = 0 \ or \ 1 \qquad (5.9)$$

補助問題５．１．２における決定変数の表示法は問題５．１と同じである。補助問題５．１．１と補助問題５．１．２が同じ問題を表わしていることは明らかである。解の目的関数値の間には式（５．10）の関係があり、解は互いに変換できる。

$$F_1 + F_2 = \sum_k f_k \qquad (5.10)$$

多次元ナップザック問題の近似解法には、千住・豊田の有効勾配（effective gradient）を使う方法[2)3)]を採用する。これは、補助問題５．１．２の形の問題を対象とし、主実行可能な方法と双対実行可能な方法がある。ここでは、分枝変数の決定に補助問題の計算過程を利用できる双対実行可能な方法を用いる。この方法での有効勾配は、制約式が資源制約、変数がプロジェクトの採否を表わすとしたとき、次のことを表わす。すなわち、計算段階での超過資源ベクトルに各プロジェクトの消費資源ベクトルを投影した長さに対する目的関数の係数の割合のことである。有効勾配が最小のプロジェクトを捨て、これを実行可能解に達するまで続けるのである。

連結網化分枝と実行可能化分枝を用いる問題５.１の厳密解法のアルゴリズムは次のようになる。

ステップ１：各リンクを除くことによる目的関数の増加の下限値 f_k を求める。

ステップ２：近似解法によって初期解 $\boldsymbol{x}^* = \{x_1^*, x_2^*, \cdots, x_k^*, \cdots\}$ を求め、その目的関数を Z^* とする。

ステップ３：すべての変数を自由変数、すなわち $x_k \leftarrow -1$ とする。

ステップ４：節点の表わす解集合の中での最短連結網を求める。最短連結網が求まればステップ９へ、連結網がなければステップ５へ進む。

ステップ５：その節点への分枝変数 x_k の値が１ならばステップ６へ、０ならばステップ７へ進む。

ステップ６：ひとつ前の節点へあともどりして $x_k \leftarrow 0$ とし、もう一方の節点へ分枝する。ステップ４へもどる。

ステップ７：$x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。

ステップ８：その節点が探索を開始した節点ならば計算を終える。そうでなければステップ５へもどる。

ステップ９：最短連結網を構成するリンクで $x_k=-1$ のものをリンク長の短い順に分枝変数に選び、 $x_k\leftarrow 1$ として分枝する。ただし、分枝により制約式を満たすことができなくなれば、そこで分枝を打ち切りステップ６へもどる。そうでなければステップ10へ進む。

ステップ10： $x_k=0$ あるいは $x_k=1$ の固定変数を制約に加え、補助問題５．１．２を解く。近似解が求まればステップ11へ、実行可能解がなければステップ14へ進む。

ステップ11：補助問題を解くときに、有効勾配が小さいとして選択された順に分枝変数に選び、 $x_k\leftarrow 0$ として実行可能解まで分枝する。ただし、分枝により、その節点において $x_k=0$ であるリンクの f_k の合計値と最大ネットワークの目的関数値の和が Z^* を超えたならば、そこで分枝を打ち切りステップ16へ進む。そうでなければステップ12へ進む。

ステップ12： $x_k\neq 0$ であるリンクからなるネットワークの目的関数値　　を求める。

ステップ13： $Z<Z^*$ ならば Z^* を Z で、$\boldsymbol{x}^*$ を $\boldsymbol{x}=\{|x_k|\}$ で置き換える。ステップ14へ進む。

ステップ14：その節点への分枝が連結網化分枝によるものならばステップ５へもどる。そうでなければステップ15へ進む。

ステップ15：その節点への分枝変数 x_k の値が０ならばステップ16へ、１ならばステップ17へ進む。

ステップ16：ひとつ前の節点へあともどりして $x_k\leftarrow 1$ とし、もう一方の節点へ分枝する。ステップ10へもどる。

ステップ17： $x_k\leftarrow -1$ とし、ひとつ前の節点へあともどりする。ステップ14へもどる。

計算を終えたときの $\boldsymbol{x}^*$ が最適解である。ステップ２で初期解を求める近似解法には backward 法を用いる。これが基本問題に対する backward 法と異なるのは、複数の制約式のすべてが満たされるかどうかで解の実行可能性を調べることである。また、ここでは連結網化分枝は最短連結網を構成するようにした。だが、たとえば予算制約がある場合には最小建設費用の連結網を求め、予算内で連結網をつくれるかどうかの検査をするように直せばよい。

ここに示したアルゴリズムでは補助問題を近似解法で解いている。そのため、目的関数の下限値を使って早い段階で分枝を打ち切れない。もし最適解を求めたならば、問題５．１の目的関数の下限値は式（５.11）で求まる。

$$Z^l=Z^0+\sum_k f_k-F_2 \qquad (5.11)$$

ここに、Z^l：目的関数の下限値

Z^0：最大ネットワークの目的関数値

しかし、補助問題５．１．２を使って求めた目的関数の下限値が強力でないのは、基本問題の例から明らかである。そこで、補助問題の近似解の目的関数値を式（５.11）に代入し、得られる Z^l を下限値のように扱っても最適解を逃すことはほとんどないと考えられる。すなわち、解の精度が極めて高い近似解法になる。

近似解法のアルゴリズムは、厳密解法のステップ10以降を次のように変更すればよい。

ステップ10： $x_k=0$ あるいは $x_k=1$ の固定変数を制約に加え、補助問題５．１．２を解く。

近似解が求まればステップ11へ、実行可能解がなければステップ16へ進む。

ステップ11：補助問題の近似解の目的関数値 F_2 から目的関数の下限値 Z^l を計算する。

ステップ12：$Z^l \leqq Z^*$ ならばステップ13へ、$Z^l > Z^*$ ならばステップ16へ進む。

ステップ13：補助問題を解くときに、有効勾配が小さいとして選択された順に分枝変数に選び、$x_k \leftarrow 0$ として実行可能解まで分枝する。

ステップ14：$x_k \neq 0$ であるリンクからなるネットワークの目的関数値 Z を求める。

ステップ15：$Z < Z^*$ ならば Z^* を Z で、$\boldsymbol{x}^*$ を $\boldsymbol{x}=\{|x_k|\}$ で置き換える。ステップ16へ進む。

ステップ16：その節点への分枝が連結網化分枝によるものならばステップ5へもどる。そうでなければステップ17へ進む。

ステップ17：その節点への分枝変数 x_k の値が0ならばステップ18へ、1ならばステップ19へ進む。

ステップ18：ひとつ前の節点へあともどりして $x_k \leftarrow 1$ とし、もう一方の節点へ分枝する。ステップ10へもどる。

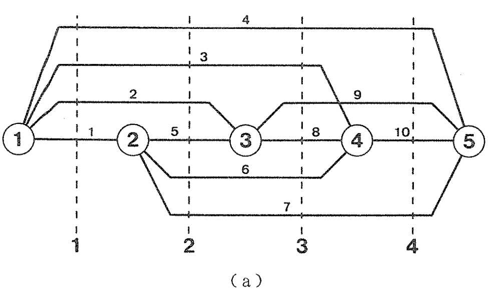
ステップ19：$x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。ステップ16へもどる。

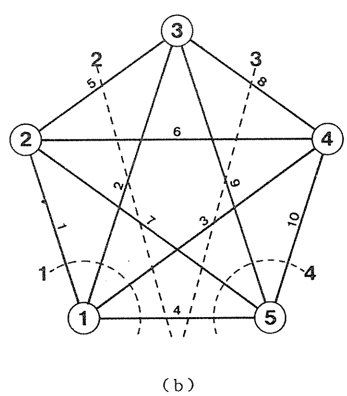
5．2．4　複数制約条件の問題の計算例[1)]

ここでは、多次元ナップザック問題と同じ形の制約式を持つ最適交通網構成問題の具体的な例題に、5．2．3で提案した解法を適用する。

問題5．1の制約条件はリンクの組み合わせに関するものだけで、しかもリンクを除けば必ず実行可能解に近づくものであった。このような制約条件をもつ最適交通網構成問題の例に、列車やバスの急行運転の運行パターンを決める問題がある。ここでは、この問題を例題として取り上げる。

列車の運行系統数の制約下で、乗客の総所要時間を最小にするように運行パターンを決めるものとする。5駅からなる路線を例に、すべての運行パターンを示すと図－5．1（a）のようになる。この図では停車駅をノード、停車駅間をリンクで表わしている。そして、これを書き直せば、図－5．1（b）のように一般的なネットワークの形になる。運行系統数が制約条件として与えられれば、各駅間を走行する列車の種類がその値以下でなければならない。したがって、隣接駅間の断面を表わすカットを設け、カットに含まれるリンク数を制約すればよい。カットの例





図―5.1　運行パターンのネットワーク表示

は、図－5．1に示すとおりである。

列車の通過待ちや乗換時間などは考えないものとすると、運行パターンを決める問題は問題5．1と同じ形で表わすことができる。すなわち、列車の運行パターンを最適化する問題は次のように定式化できる。

問題5．2

$$\min \quad Z=\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij} t_{ij}(\boldsymbol{x}) \qquad (5.12)$$

$$\text{s.t.} \quad \sum_{k=1}^{m} a_{kl} x_k \leqq M_l \quad (l=1,2,\cdots,n-1) \qquad (5.13)$$

$$x_k = 0 \ or \ 1 \quad (k=1,2,\cdots,m) \qquad (5.14)$$

ここに、Z：総所要時間

t_{ij}：ノード ij 間の最短所要時間

a_{kl}：リンクがどのカットに含まれるかを表わし、リンク k がカット l に含まれるとき1、含まれないとき0として与える。

M_l：カット l に含まれるリンク数の上限値で、計画運行系統数を表わす。

m：リンク数

n：ノード数

式（5．13）の制約式で、各駅間に走らせる列車の種類数を定める。たとえば、図－5．1の場合に全区間で2種類にすると、制約式は次のようになる。

$$x_1+x_2+x_3+x_4 \leqq 2 \qquad (5.15)$$

$$x_2+x_3+x_4+x_5+x_6+x_7 \leqq 2 \qquad (5.16)$$

$$x_3+x_4+x_6+x_7+x_8+x_9 \leqq 2 \qquad (5.17)$$

$$x_4+x_7+x_9+x_{10} \leqq 2 \qquad (5.18)$$

式（5．15）は駅1と駅2の間で列車を2種類以下にする制約式である。式（5．16）～（5．18）についても同様である。

例題5．1として阪急電鉄神戸線の梅田－西宮北口間を取り上げ、実際のデータを用いて計算を行う。駅数は8で、需要交通量は表－5．1に示すものとする。これは、昭和50年の交通セン

表－5.1　例題5.1の需要交通量

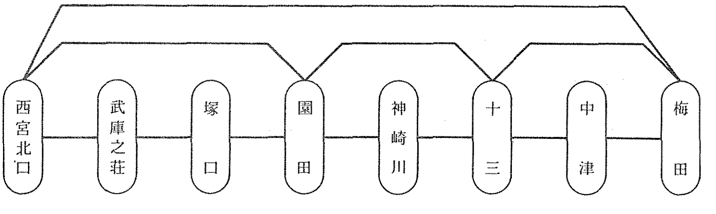
発駅	着駅 梅田	中津	十三	神崎川	園田	塚口	武庫之荘	西宮北口	合計
梅田	0	102	715	551	433	1033	138	2914	5886
中津	56	0	60	12	13	37	0	66	244
十三	460	319	0	820	402	1260	113	4399	7773
神崎川	1623	89	831	0	30	134	12	186	2905
園田	3313	149	818	107	0	309	68	872	5636
塚口	8215	216	2012	151	207	0	112	3344	14257
武庫之荘	3324	132	674	55	127	325	0	1800	6437
西宮北口	19537	389	3426	181	437	2286	476	0	26732
合計	36528	1396	8536	1877	1649	5384	919	13581	69870

サスの結果から求めた午前８時から９時30分までの駅間OD表である。対象区間外からの流入、対象区間外への流出を補正したものである。リンクの長さになる停車駅間の所要時間は表－５．２に示す。

表—5.2　例題5.1の停車駅間所要時間　（単位：s）

起点 ＼ 終点	梅田	中津	十三	神崎川	園田	塚口	武庫之荘	西宮北口
梅田		170	285	355	505	680	755	890
中津	170		150	235	390	530	610	840
十三	285	150		160	295	470	540	750
神崎川	355	235	160		230	370	440	620
園田	505	390	295	230		220	295	470
塚口	680	530	470	370	220		170	370
武庫之荘	755	610	540	440	295	170		240
西宮北口	890	840	750	620	470	370	240	

計画運行系統数を全区間で３種類とし、５．２．３で提案した近似解法を適用した。その結果、図－５．２に示す解が得られた。解に含まれるリンクを組み合わせて、どのような列車にするかはこの問題で扱っていない。しかし、図－５．２からは梅田から西宮北口まで停車しない特急と、十三、園田に停車する急行、それに各駅停車とする案が考えられる。実際には特急も十三に停車し、急行は園田でなく塚口に停車している。得られた解での乗客の平均所要時間は11.5分で、現行の12.2分よりも5.5％短い。



図—5.2　例題5.1の解の運行パターン

このように、５．２．３で提案した解法は、実際的な最適交通網構成問題への適用が可能である。また、規模の小さい問題を厳密解法と近似解法で解いた結果では、近似解法で得られた解は最適解であった。

５．２．５　むすび

この節では、複数の制約条件を持つ最適交通網構成問題の解法について考察した。制約式を分類し、それらを組み合わせてできる複数制約条件の問題に対して解法をどのようにすればよいかを述べた。そして、比較的容易に解を求められる、多次元ナップザック問題と同じ制約式を持つ問題の解法を提案した。これは、多次元ナップザック問題の近似解法を使って実行可能化分枝を行う厳密解法で、近似解法に直したものも提案した。この近似解法は具体的な例題に適用し、十分に実用が可能であることを確めた。

5.3 多目的計画問題への拡張

5.3.1 はじめに

この節では、複数の目的関数を持つ最適交通網構成問題について考察する。まず交通網計画において、どのような場合に複数の目的関数が必要となるかを述べ、それに対する従来の研究を示す。つぎに、最適交通網構成問題を多目的化した問題の特徴を明らかにし、どのように解法を構成すればよいか検討する。そして、目的関数のスカラ化手法を用いる多目的最適交通網構成問題の解法を提案する。

5.3.2では、交通網計画における多目的計画問題の必要性と、多目的の交通網計画問題に関する従来の研究について述べる。

5.3.3では、多目的の最適交通網構成問題が通常の多目的決定問題と異なる点を明らかにし、解法の基礎的な考察を行う。

5.3.4では、多目的最適交通網構成問題の解法を一般的な形で示す。

5.3.2 交通網計画における多目的計画問題の必要性と従来の研究

ここでは交通網計画において、どのような場合に複数の目的関数を扱う必要があるのか述べる。そして、多目的の交通網計画問題に関する従来の研究について述べる。

交通網計画のなかで、もっとも多く行われるのは道路網計画あるいは街路網計画である。道路網計画の最大の目的は、需要交通量を円滑に流せる道路網をつくることである。そのため、目的関数には総走行距離や総走行時間の最小化が採用される。しかし、交通量の多い道路区間では沿道に及ぼす騒音や排出ガスの影響が問題にされるようになり、道路網計画の策定に際してはこれらの環境要因を考慮することが必要になった。すなわち、このような場合には道路網の本来の機能の評価値である総走行時間とともに、騒音や排出ガスが道路周辺へ及ぼす影響を表わす指標をも目的関数にしなければならない。

環境要因だけでなく、道路網に多くの機能を要求する場合には複数の目的関数が必要である。道路網に対する要求が制約条件にするほど強いものではないとき、それを目的関数にして弱い制約を課すことがよくある。たとえば、総走行距離が短い道路網を求めるが、ある特定のノード間の最短距離が長くならないようにしたいという場合が考えられる。さらに、総走行距離か総走行時間かというように、類似の目的関数の選択が困難な場合も考えられよう。そして、これらのケースが重なることもある。

交通網計画を対象に複数の目的関数を扱った例としては、大根田・渡辺・森地の街路網計画の研究、[4] 日野・西村の最適ネットワーク構成の研究、[5] 吉川・小林らの道路網整備の研究[6]がある。森地らの研究は建設費と総走行距離、基準騒音レベル超過面積を目的関数とし、伏見・山口の複数の目標をバランスよく達成させるための数理計画的方法[7]を用いている。しかし、決定変数はリンク交通量であり、本来の最適交通網構成問題とは異なる。むしろ、交通配分問題であるといえる。吉川らは目標計画法を用いているが、これも交通配分問題である。これに対し、西村らは総走行距離と建設費を目的関数とし、伏見・山口の方法と加重和最小化法を最適ネットワーク構成問題に適用した簡単な例を示している。けれども、解法の詳細は述べられていない。

このように、交通網計画において複数の目的関数を扱う必要があることは、すでに認識されて

いる。だが、最適交通網構成問題としての本格的な取り組みはなされていない。それは、解を求めるのに必要な計算量が単一目的の場合よりもいっそう大きくなり、多目的化が必要な実際的な問題への適用が困難であると考えられたからであろう。多目的システムの最適化理論が実際的な問題に頻繁に使われるように一般化していないことも、応用が図られない一因であろう。しかし、複数の目的関数を用いれば現実に則した問題の定式化が可能になり、その意義は大きい。

5．3．3　最適交通網構成問題の多目的化

ここでは、最適交通網構成問題を多目的化したとき、通常の多目的決定問題と異なる点を明らかにする。そして、どのように解法を構成すればよいか、基礎的な考察を行う。すなわち、従来の多目的システムの最適化手法で適用できそうなものはどれか、どのような改良が適用に際して必要かを検討する。

多目的計画問題は複数の目的関数をもつ最適化問題で、その解は非劣解とよばれている。複数の目的関数のすべてを最適にする解が得られないのが普通であり、そのときは複数の非劣解が求まる。唯一の解を求めることが必要ならば、非劣解集合のなかから特定の解を選び出さなければならない。この多目的計画問題の非劣解集合から選好最適解を求める問題は多目的決定問題とよばれる。[8)] 最適交通網構成問題で最終的に求めたいのは、ひとつの交通網案である。したがって、目的関数を複数にすれば、多目的決定問題を解くことになる。

多目的決定問題では、意志決定者の選好目的関数に基づいて選好最適解が求められる。この選好最適化の方法の主なものはふたつに分けられる。ひとつは選好目的関数を仮定する目的関数のスカラ化手法である。目的関数はすべて最小化であるとして、これには加重和最小化法、最大成分最小化法、ϵ －制約式法などがある。もうひとつは決定者の局所的選好に基づき、逐次的に選好最適解に到達する対話型最適化手法である。これには対話型 Frank－Wolfe 法や SWT 法（surrogate worth trade－off method）などがある。[8)]

これらの最適化手法を最適交通網構成問題に適用しようとするとき、対象とする問題の特徴を把握しておかなければならない。とくに重要なのは、次の２点である。

①　０－１整数計画問題である。

②　目的関数か制約式に最短距離などのネットワークの関数値を含む。

通常の多目的システムの最適化理論は連続変数を対象にしたものであり、０－１整数計画問題を扱うことを目的にしたものはない。多目的決定問題の手法を適用しようとするとき、この点が最大の障害となる。選好最適化手法には、目的関数が連続で微分可能であることを前提としているものがある。だが、このような手法は最適交通網構成問題には適用できない。この問題では制約式を満たす解の集まりは連続な実行可能領域でなく、離散的な実行可能解の集合で与えられるからである。したがって、実行可能解集合に属する解を列挙して調べてゆく場合にも適用できる手法でなければ、最適交通網構成問題には使えない。

選好最適化手法のうちの目的関数のスカラ化手法を用いれば、単一目的関数の計画問題を解くことになる。一般の多目的計画問題では非線形計画問題になるが、ここでの問題では単一目的の最適交通網構成問題になる。そこで、目的関数によっては解を得るのが困難な場合も考えられるが、スカラ化手法は最適交通網構成問題への適用が可能である。

対話型最適化手法は限界代替率やトレードオフ比、代用価値関数などを用い、非線形計画法の手法で解を順次、改良してゆくものである。最適交通網構成問題では目的関数値は解ごとに連続的に変化するのでなく、離散的な値しかとれない。しかも、決定変数の値によってその値が単純に変化する目的関数を用いることは少ない。そのため、目的関数値で解を改良する方向を決めても、それに対応する解は必ずしも存在しない。そして、変数のとり得る値は0か1しかなく、その一方で変数の数は多いという構造であり、局所という概念も異なる。したがって、対話型最適化手法の場合には非線形計画法の部分を置き換えるとしても、最適交通網構成問題には適用できない。基本的な考え方をもとに、最適交通網構成問題に適した手法を新たに作らなければならないであろう。

適用が可能な目的関数のスカラ化手法を用い、多目的の最適交通網構成問題の解法を作成するとき、これらの手法では繰り返し計算が多いことに注意しなければならない。たとえば、加重和最小化法では目的関数の重みを変えて解を求め直すことが行われる。また ε－制約式法でも、制約式化した目的関数の許容範囲である ε をパラメトリックに変えて解く。この場合、同じ計算が繰り返されることが多い。最適交通網構成問題では解を求めるのに必要な計算量が大きいので、むだな繰り返し計算は避けなければならない。そこで、前回の計算結果を利用するとか、あらかじめ非劣解を列挙しておくなどの対策がいる。

目的関数を複数にした最適交通網構成問題を考えなければならないとき、対象とするネットワークの規模は大きいであろう。そして、単一目的関数の問題を解く部分には近似解法が使われると考えられる。このような状態では、選好最適解を求める部分も厳密なものでなくてもよい。解法全体のバランスがとれていれば、選好最適化も近似解法でよいであろう。そうすれば、選好解を求めるための計算時間を短縮できる。

多目的の最適交通網構成問題と、その解法に関する基礎的な考察をまとめると、次のようになる。

① 最適交通網構成問題は0－1整数計画問題であり、この点で従来の多目的システムの最適化理論の扱う問題と異なる。

② 目的関数をスカラ化する手法は最適交通網構成問題に適用できる。非線形計画問題のかわりに単一目的の最適交通網構成問題を解けばよい。

③ 対話型最適化手法の適用は困難である。基本的な考え方をもとに、最適交通網構成問題に適した手法を作らなければならない。

④ 目的関数のスカラ化手法を適用する場合、繰り返し計算の必要性が大きいことへの対策を考えなければならない。

⑤ 単一目的の最適交通網構成問題の解法には近似解法を使うことになる。そこで、選好最適化にも近似解法の使用を考えてよい。

5.3.4 多目的最適交通網構成問題の解法の提案[9)]

ここでは、5.3.3での考察に基づき、目的関数のスカラ化手法を用いる多目的最適交通網構成問題の解法を提案する。解法の細部は目的関数や使用する選好最適化の手法により異なる。そこで、多目的の場合に共通する一般的な形で解法を示す。

解法の作成の対象とするのは、次の多目的最適交通網構成問題である。

問題５．３

$$\min \quad \boldsymbol{f}(\boldsymbol{x}) \tag{5.19}$$

$$\text{s.t.} \quad \sum_k c_k x_k \leqq C \tag{5.20}$$

$$x_k = 0 \ or \ 1 \tag{5.21}$$

ここに、$\boldsymbol{f}(\boldsymbol{x})=\{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_i(\boldsymbol{x}), \cdots\}$

：$f_i(\boldsymbol{x})$ を成分とするベクトル目的関数

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_k, \cdots\}$

：ネットワークの状態を表わし、リンク k がネットワークに含まれないとき $x_k=0$ 、ネットワークに含まれるとき $x_k=1$ とする。

c_k：リンク k の建設費

C：ネットワークの建設費の制約値

式（５．19）は複数の目的関数を成分にもつベクトル目的関数を最小化することを表わす。式（５．20）はネットワーク建設費の予算制約である。すなわち、問題５．３は予算制約下で複数の目的関数を最小化するものである。予算制約はもっともよく使われる制約条件であり、実際的な問題のほとんどに含まれると考えられる。複雑な制約条件を扱うことはこの節の目的ではないので、制約条件は予算制約だけの問題を対象とする。

５．３．３での考察より、選好最適化の手法には目的関数をスカラ化するものを用いる。この場合、目的関数の重みなどをパラメリックに変えて行う繰り返し計算への対策が必要であった。そして、選好最適化も近似解法でよいのではないかということであった。これらを考慮して多目的の場合の選好解を求める方法を考える。

まず、繰り返し計算への対策を考える。多目的決定問題は非劣解集合の中から選好最適解を選び出すものである。繰り返し計算を行っても、もとになる非劣解集合は同じである。目的関数の重みや許容範囲が変化しても、選好最適解の候補である非劣解は変わらない。よって、あらかじめすべての非劣解を含む非劣解集合を求めておけば、最初から解を調べ直さなくてよい。そこで、非劣解集合の作成を検討する。

ある解 $\boldsymbol{x}^*$ が非劣解であるということは、次の式（５．22）を満たす解 $\boldsymbol{x}$ が存在しないことである。

$$\boldsymbol{f}(\boldsymbol{x}) \leqq \boldsymbol{f}(\boldsymbol{x}^*) \tag{5.22}$$

この式は、すべての i に対して $f_i(\boldsymbol{x}) \leqq f_i(\boldsymbol{x}^*)$ で、かつ少なくともひとつの i に対しては $f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*)$ であることを表わす。つまり、ある解が非劣解であるとするには、式（５．22）を満たす解がないことを確めなければならない。したがって、非劣解を探し出すには実行可能解を列挙し、それらの２個の解について式（５．22）の関係を調べ、非劣解でないものを除く操作を繰り返す必要がある。

このように、すべての非劣解を求めるには、基本的にはあらゆる実行可能解の目的関数値を計算しなくてはならない。実際的な問題では実行可能解は多く、それらの目的関数値をすべて計算

することは不可能といってよい。そこで、完全な非劣解集合を作成するのでなく、一部分だけを含む部分集合を作るか、それに代わるものを作るしかない。しかし、非劣解かどうかを調べるにはすべての実行可能解が必要なので、部分集合を作る方法もとれない。よって、完全な非劣解集合に代わるものを作成する方法だけが残る。

すべての実行可能解を列挙するのが不可能ならば、選好される可能性の高い実行可能解だけを列挙することが考えられる。そして、列挙した実行可能解だけを対象に、非劣解を探し出せばよい。この方法で作った非劣解集合には、全実行可能解に対する真の非劣解がすべて含まれる保証はない。そのかわりに、列挙した実行可能解に対してのみ成り立つ非劣解が含まれる可能性がある。したがって、選好されるかもしれない解を検討対象から除くことになり、選好最適化を近似解法にすることになる。しかし、この方法は最適交通網構成問題で近似解法の使用が認められてきたように、実用上は許される近似化であると考える。

選好最適化を近似解法にする方法はほかにもある。目的関数のスカラ化手法を用いれば、単一目的関数の最適交通網構成問題ができる。これを近似解法で解くと、選好最適化に近似解法を使用することになる。ひとつの選好解を求めるだけのときは、この方法が有効である。しかし、繰り返し計算を行うときは、むだな計算を省くのに非劣解を列挙しておく方法が優れている。

つぎに、どのような範囲で実行可能解を列挙し、非劣解集合を作成するかを考えなくてはならない。選好最適解の選択方法はいろいろあるが、ある目的関数についてみたとき、その最小値から大きく隔った値をもつ解は、非劣解であっても選好されることが少ないと考えられる。あるいは、その解の目的関数値よりも小さい値をもつ解が多数あれば、選好されることが少ないであろう。そこで、あるひとつの目的関数だけを考えて実行可能解を列挙してゆき、目的関数値が最小値からある範囲内であり、上位から一定の順位以内の解だけを対象に、非劣解を探す操作を行うものとする。

この方法においても、列挙される実行可能解が多くなりすぎないようにしなければならない。そこで、**4.3** で解の探索範囲をある近似解の近傍に限定したのと類似の方法を考える。すなわち、あるひとつの目的関数を対象に近似解 $\boldsymbol{x}$ を求めたとする。そして、その近似解に含まれるリンクのうち、ネットワークから除いても目的関数の増加量が小さいものと、近似解には含まれないが、ネットワークにつけ加えれば目的関数の減少量が大きいリンクだけを検討リンクとする。残りのリンクは固定リンクとし、検討リンクの組み合わせでできる範囲に解の探索を限定する。目的関数は複雑なので、次の式（5.23）でリンクの重要さを表わす評価値を定め、これにより検討リンクを決めるものとする。

$$F_k = \frac{1}{c_k} \sum_i \alpha_i \{ f_i(\boldsymbol{x}_k) - f_i(\boldsymbol{x}) \} \qquad (5.23)$$

ここに、F_k：近似解でリンク k の状態を変えたときの目的関数値の変化の大きさを表わす。

α_i：目的関数のスケールを調整する係数

$\boldsymbol{x}_k$：$\boldsymbol{x}$ のネットワークにおいて、リンク k の状態を逆に変えたネットワークを表わす。

リンクの評価値 F_k は、リンク k が近似解のネットワークに含まれるとき、一般に正の値とな

る。その値が大きいほど、リンクkをネットワークから除いたときの目的関数値の増加量が大きい。反対に、リンクkが近似解に含まれないとき、F_kは一般に負の値となる。その絶対値が大きければ、リンクkをネットワークに付け加えたときの目的関数値の減少量が大きいことになる。そこで、近似解に含まれるリンク、含まれないリンクのそれぞれについて、F_kの小さい順に一定数のリンクを検討リンクとする。

多目的最適交通網構成問題の解を求める手順をまとめると、次のようになる。

ステップ1：あるひとつの目的関数を対象に近似解を求める。

ステップ2：近似解のネットワークで各リンクの状態を逆にしたときの目的関数値の変化量を計算する。

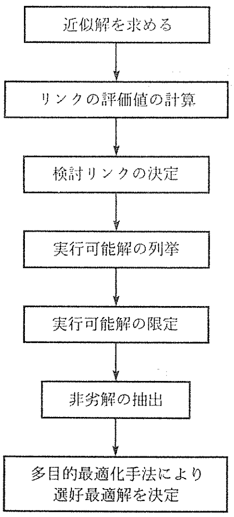
ステップ3：式（5.23）でリンクの評価値を計算し、検討リンクを決める。

ステップ4：検討リンクの組み合わせにより実行可能解を列挙する。

ステップ5：ある目的関数について、その最小値からの隔たり、あるいは小さい順に並べた順位により実行可能解を限定する。

ステップ6：実行可能解の中から非劣解を探し出す。

ステップ7：目的関数をスカラ化する多目的決定問題の最適化手法を適用し、選好最適解を決定する。



図—5.3　多目的最適交通網構成問題の計算手順

計算手順を図示したのが**図－5.3**である。ステップ1での目的関数はどれを選んでもよい。だが、計算時間を短くするためには最短経路配分を行っての総走行距離などの計算が簡単なものがよい。これはステップ5で使用する目的関数についてもあてはまる。通常はステップ1とステップ5で用いる目的関数は同じものでよい。そして、ステップ4～6は、ひとつのステップが終了してから次に進むのでなく、可能ならば並行して実行すればよい。ステップ7では加重和最小化法やε－制約式法などを適用し、選好最適解を求める。目的関数の重みなどを変えて解き直してもよいし、必要であれば複数の手法による解の比較をすればよい。このようにしたとき、最終的にどの解を選ぶかは意志決定者の判断による。

ここでは、多目的最適交通網構成問題の解を得るための計算手順を一般的な形で示した。実際の具体的な問題に適用するには、目的関数や選好最適化の手法に応じて解法の詳細な部分を作らなければならない。このとき、目的関数のトレードオフが顕著であれば、実行可能解の限定に注意がいる。ここに示した方法では選択した目的関数の値が極めて大きい解は除かれ、他の目的関数では小さい値を持つ解がすべて非劣解を選ぶ対象からはずされることが起こり得るからである。

5.3.5 むすび

この節では、多目的最適交通網構成問題について考察した。最初に、交通網計画においてどのようなときに複数の目的関数を扱わなければならないかを述べた。つぎに、多目的最適交通網構成問題と、従来の多目的システムの最適化理論が対象としている問題との相違を明らかにした。そして問題の特徴を考慮し、それに適した解法を提案した。

ここで提案した解法は、目的関数を特定していない問題に対するものである。実際に適用するときには、それぞれの問題の特殊性に応じて解法の修正が必要となるかもしれない。そして、最適交通網構成問題のような離散的な決定変数を持つ問題に対する多目的最適化の手法の開発が今後の課題である。

5.4 交通網の段階建設問題への拡張[10)]

5.4.1 はじめに

この節では交通網を段階的に建設する問題を取り上げ、最適交通網構成手法による解法を提案する。

大規模な交通網は、その完成までに長期間を要するのが普通である。この場合、完成後だけでなく建設途中における需要交通量と交通網の関係を考慮し、どのように交通網をつくってゆくかを決めなければならない。まず、この問題を最適交通網構成問題として定式化し、その解法を提案する。そして、簡単な計算例を示す。

5.4.2では、交通網の段階建設問題について述べ、最適交通網構成問題として定式化する。

5.4.3では、分枝後退法を用いる交通網の段階建設問題の解法を提案する。

5.4.4では、交通網の段階建設問題の計算例を示す。

5.4.2 交通網の段階建設問題

交通網を計画するとき、よく使われる方法は需要交通量を設定し、それに対して適切な交通網案を定めるものである。需要交通量を固定すれば、最適交通網構成手法もこれと同じ方法である。しかし、交通網の建設が長期間にわたる場合、建設途中の交通網でもそのときの交通需要をさばかなければならない。この場合、完成時点の需要交通量に対して最適な交通網を求めるのでなく、建設途中の需要交通量を含めた全期間に対しての最適化を図る必要がある。そして、最終的な交通網案だけでなく、どのような順序で交通網を構成してゆくかを決めなければならない。そこで、ここでは建設期間をいくつかに分け、各期間の需要交通量を与え、どのリンクをどの期間につくるかを決める問題を考える。

段階建設問題の基本的なものとして、ここでは建設費の予算制約下で総走行距離を最小化する問題を対象とする。各期間ごとに総走行距離を最小化する目的関数があるとすれば、この問題は多目的決定問題になる。しかし、このような場合の一般的な扱い方に従い、期間ごとの総走行距離の加重和の最小化を目的関数とする。そして、多目的最適交通網構成問題としては扱わない。

段階建設問題の前提条件として、次の仮定を設ける。

① 需要交通量と予算は各期間ごとに与えられる。

② 目的関数における各期間の重みはあらかじめ与えられる。

③　各期間の予算が余っても、次の期間に繰り越さない。

④　１本のリンクを複数期間にわたって建設することはしない。

⑤　建設されたリンクがそれ以後の期間に除去されることはない。

⑥　ある期間に建設したリンクは次の期間以後に使用できる。

これをもとに問題を定式化すると、次のようになる。

問題５．４

$$\min \quad Z=\sum_{t=1}^{T}\{w^t\sum_{i=1}^{n}\sum_{j=1}^{n}q_{ij}^t\,d_{ij}^t(\boldsymbol{x}^t)\} \tag{5.24}$$

$$\text{s.t.} \quad \sum_{k=1}^{m}c_k(x_k^t-x_k^{t-1})\leqq C^t \quad (t=1,\ 2,\cdots,T) \tag{5.25}$$

$$x_k^t-x_k^{t-1}\geqq 0 \quad (k=1,\ 2,\cdots,m,\ t=1,\ 2,\cdots,T) \tag{5.26}$$

$$x_k^0=0 \quad (k=1,\ 2,\cdots,m) \tag{5.27}$$

$$x_k^t=0\ or\ 1 \quad (k=1,\ 2,\cdots,\ m,\ t=1,\ 2,\cdots,T) \tag{5.28}$$

ここに、Z：各期間の総走行距離の加重和

w^t：期間 t 終了時のネットワークに対する重み

q_{ij}^t：期間 t 終了時のネットワークに対するノード ij 間の需要交通量

d_{ij}^t：期間 t 終了時のネットワークにおけるノード ij 間の最短距離

$\boldsymbol{x}^t=\{x_1^t,x_2^t,\cdots,x_k^t,\cdots,x_m^t\}$

：期間 t 終了時のネットワークの状態を表わす。リンク k が期間 t 終了時のネットワークに含まれるとき $x_k^t=1$ 、含まれないとき $x_k^t=0$ とする。

c_k：リンク k の建設費

C^t：期間 t の予算

m：リンク数

n：ノード数

T：期間数

式（５.24）の目的関数は、各期間の総走行距離に重みをかけ、全期間について合計したものを最小化することを表わしている。建設途中のネットワークも、それぞれの重みで目的関数において考慮される。式（５.25）は予算制約である。各期間に建設されるリンクの建設費の合計が、その期間の予算に納まらなければならないことを表わしている。この予算は少なくとも需要交通量のあるノード間に経路を設けられる額が必要である。式（５.26）は、建設されたリンクはそれ以後の期間のネットワークに必ず含まれることを表わしている。式（５.27）は期間０、すなわち建設開始時のリンクの状態を表わす。既存リンクとして建設開始時に存在するリンクを扱うときは、そのリンクに対して $x_k^0=1$ とすればよい。

この問題の特徴は次のとおりである。

①　リンクをどの期間につくるか、あるいはつくらないかを決める。

②　目的関数においては、複数のネットワークを考慮しなければならない。

③　期間ごとに個別にネットワークを決められない。全期間に対する最適化を行う。

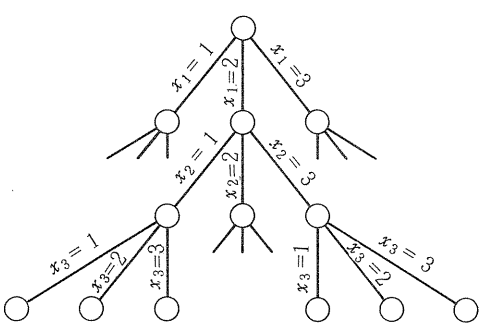
5．4．3　交通網の段階建設問題の解法

ここでは、5．4．2で定式化した交通網の段階建設問題の解法を考える。段階的に建設しようとする場合はネットワークの規模が大きいのが普通である。そのため、実用には近似解法が必要である。しかし、ここでは基礎的な考察を目的とし、問題に対する理解を深めるために厳密解法を作成する。

多段階決定問題でよく使われる手法は動的計画法である。問題5．4に動的計画法を適用するならば、まず期間1の実行可能解のなかで、最適解の一部になる可能性のある解を列挙する。つぎに、列挙した解のそれぞれに期間2の予算制約内のリンクを付加し、期間2までに対する解で最適解の一部になる可能性のある解を列挙する。そして、この操作を繰り返せばよい。

動的計画法を用いる方法では、ネットワークの規模が大きくなると、各期間で保持しなければならない解の数が急激に増大する。そのため、極めて小規模のネットワークにしか使えないであろう。また、同じネットワークの目的関数値の計算を繰り返すことが多いと考えられる。したがって、動的計画法は交通網の段階建設問題を解くのには適していない。そこで、通常の最適交通網構成問題と同じように分枝後退法を使用する。

問題5．4の決定変数は0－1変数である。1本のリンクの状態はT個の変数で表わされる。よって、変数の定義からはリンクの取り得る状態の数は2^Tである。しかし、式（5．26）の制約により、実際には（$T+1$）しかない。つまり、どの期間につくるのか、あるいはつくらないかである。そこで、解法を考えやすくするために決定変数をリンクをつくる期間で表わす。すなわち、リンクkを期間tにつくるとき$x_k=t$とする。最終期間Tまでにつくらないときは、$x_k=T+1$で表わす。このようにすれば、解を系統的に列挙するための探索木は、たとえば図－5．4のように表わされる。この探索木は、節点から3本以上の枝が分かれることが、これまでのものと異なる。解の探索方法も探索木の変化に応じて変更しなければならない。



図－5.4　段階建設問題の探索木の例

分枝後退法による計算手順は、最初に近似解法で初期解を求めて暫定最適解とし、分枝と後退を繰り返してこれを置き換えてゆくものである。そこで、まず初期解を求めるための簡単な近似解法を作る。対象としている問題5．4の主要な制約条件は期間ごとの予算制約である。よって、それぞれの期間のネットワークを順番に確定してゆく方法を用いればよい。問題の構造は前の期間のネットワークにリンクを付加するものなので、期間1から始めてネットワークを拡大してゆく方法を使う。

各期間のネットワークを決めるには、基本問題に対するのと同じようなforward法やbackward法が使用できる。しかし、この問題では各期間で付加されるリンクは比較的少ないので、forward法を使う方が計算時間が短いと予想される。そこでforward法を用いると、初期解を求めるた

めの近似解法のアルゴリズムは次のようになる。

ステップ1：$t \leftarrow 1$ とする。

ステップ2：初期ネットワークとして建設費が最小の生成木を求め、それを構成するリンクについて $x_k \leftarrow 1$ とする。残りのリンクについては $x_k \leftarrow T+1$ とする。

ステップ3：$x_k = T+1$ で、ネットワークに付け加えても期間 t の予算制約を超えないリンクについて、それぞれをネットワークに加えたときの部分目的関数 Z^t を求める。付け加えることのできるリンクがなければ、ステップ5へ進む。

ステップ4：ネットワークに加えたときの Z^t の値が最小のリンク k を付加する。すなわち、$x_k \leftarrow t$ とする。ステップ3へもどる。

ステップ5：$t < T$ ならば、$t \leftarrow t+1$ としてステップ3へもどる。$t = T$ ならば、ステップ6へ進む。

ステップ6：得られた解に対する目的関数値を求め、計算を終える。

ただし、Z^t は式（5.24）の目的関数における期間 t の総走行距離の部分であり、式（5.29）で表わされる。そして、変数 x_k と x_k^t の関係は式（5.30）のとおりである。

$$Z^t = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij}^t \, d_{ij}^t(\boldsymbol{x}^t) \tag{5.29}$$

$$x_k^t = \begin{cases} 0 & if \quad x_k > t \\ 1 & if \quad x_k \leqq t \end{cases} \tag{5.30}$$

この問題では、建設期間の最初のうちはあるノードを起終点とする交通需要がなく、途中の期間から交通需要が生じる場合が考えられる。したがって、ステップ2で最小建設費の生成木が期間1の予算を超えるときは、交通需要のないノードをつなぐリンクの除去を考えればよい。そして、交通需要の生じる期間までに、そのノードをつなぐようにする。

探索木での分枝方法には実行可能化分枝や連結網化分枝の考え方が応用できる。しかし、すべてのノードをつなぐ必要がない場合もあるので、実行可能化分枝だけを採用する。そうすると、簡単に実行可能解が求められなければならない。そこで、Hoang が基本問題で目的関数の下限値と分枝変数を決めるのに使ったものと類似の補助問題[11]を考える。

問題5.4を各期間ごとに見れば、基本問題とあまり変わらない。よって、各期間ごとに補助問題を作成する。定式化すると、次のようになる。

補助問題5.4.1

$$\min \quad F_1^t = \sum_{k=1}^{m} f_k^t \, y_k^t \tag{5.31}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} c_k (y_k^{t-1} - y_k^t) \leqq C^t \tag{5.32}$$

$$y_k^{t-1} - y_k^t \geqq 0 \quad (k = 1, 2, \cdots, m) \tag{5.33}$$

$$y_k^t = 0 \ or \ 1 \quad (k = 1, 2, \cdots, m) \tag{5.34}$$

ここに、F_1^t：期間 t において実行可能にすることによる総走行距離の増加の下限値

f_k^t：期間 t において、リンク k をネットワークから除くことにより生じる総走行距離の増加の下限値

y_k^t: リンク k が期間 t のネットワークに含まれるとき $y_k^t=0$ 、含まれないとき $y_k^t=1$ とする。

この補助問題は、期間 t における総走行距離の増加の下限値を最小にする解を求めるものである。式（5.32）は予算制約である。そして、式（5.33）は問題5.4の式（5.26）に対応し、前の期間までにつくられているリンクをネットワークに含めるための制約式である。リンク k が期間 $(t-1)$ のネットワークに含まれる $y_k^{t-1}=0$ のときは、式（5.33）により $y_k^t=0$ でなければならない。リンク k が期間 $(t-1)$ のネットワークに含まれない $y_k^{t-1}=1$ のときは、$y_k^t=0$ でも $y_k^t=1$ でもよい。したがって、式（5.33）の制約式は $y_k^{t-1}=0$ のリンクに対して $y_k^t=0$ とする制約式で置き換えてもよい。

補助問題5.4.1は整数計画問題なので、式（5.34）の整数制約を実数制約にゆるめた線形計画問題を解いて近似解を求める。これは、基本問題の場合と同じである。式（5.35）を満たすようにリンクを並べてあるとすれば、近似解は $y_k^{t-1}=1$ のリンクに対して式（5.36）で y_k^t の値を定めて得られる。ただし、r は式（5.37）を満たす最小の整数である。

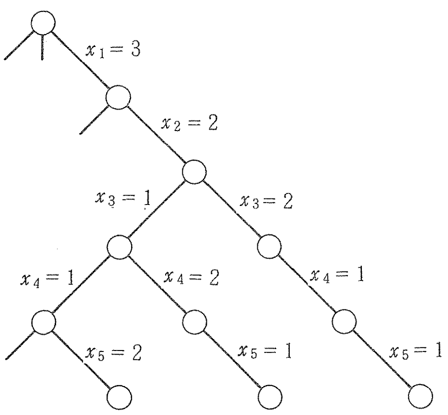
$$\frac{f_1^t}{c_1} \leqq \frac{f_2^t}{c_2} \leqq \cdots \leqq \frac{f_m^t}{c_m} \tag{5.35}$$

$$\begin{cases} y_k^t=1 & if \quad k \leqq r \\ y_k^t=0 & if \quad k > r \end{cases} \tag{5.36}$$

$$\sum_{k=1}^{r} c_k\, y_k^{t-1} \geqq \sum_{k=1}^{m} c_k\, y_k^{t-1} - C^t \tag{5.37}$$

補助問題を解くときに固定変数があれば、$y_k^t=1$ あるいは $y_k^t=0$ として制約式に加える。このときの固定変数以外の解は、式（5.37）の左辺に $y_k^t=1$ と固定したリンクの建設費をあらかじめ加えて固定変数以外のリンクで r を決め、式（5.36）を用いて求める。

この補助問題は前の期間のネットワークを前提としているので、期間1から順に解かなくてはならない。しかし、この解を使って探索木で分枝するときには $x_k=T+1$ のリンクから始め、最後に $x_k=1$ のリンクを分枝変数として実行可能解に達するようにする。そして、探索木をあともどりしたときには分枝変数を $x_k \leftarrow x_k-1$ とし、早い期間につくった場合を調べる。このような分枝方法を用いるのは、式（5.36）で近似解をつくったとき、$y_k^t=1$ としたリンクの一部を $y_k^t=0$ としても実行可能なことがあるからである。最適解を逃さないで、しかも簡単な分枝操作にするには建設期間の遅いリンクから分枝変数に選ぶのがよい。建設期間の早いものから順に分枝変数とし、あともどりしたときには遅くつくるように分枝する方法もある。このときは、実行可能解の一部のリンクを早い期間につくるように変えれば、必ず実行不可能になるようにしなければならない。そのため、分枝操作が



図―5.5　段階建設問題の解法の分枝例

複雑になる。

この分枝方法を用いると、探索木は**図－5.5**のようになる。これは期間の数を$T=2$とした例である。実行可能解に達するか、さらに分枝を続けても最適解が得られないことが明らかになれば、探索木をあともどりする。このとき、最適性の検査に補助問題5.4.1の解を使えないことに注意しなければならない。基本問題と同じようにすると、下限値を式（5.38）で求められそうに思える。

$$Z_1^l = Z^0 + \sum_{t=1}^{T} w^t F_1^t \tag{5.38}$$

ここに、Z^0: 全期間の解を最大ネットワークとしたときの目的関数値

しかし、補助問題5.4.1は期間ごとに解くため、式（5.38）で求めたZ_1^lは目的関数の下限値ではない。下限値に使うのであれば、全期間を対象とする補助問題を作らなければならない。だが、その場合は簡単に補助問題を解けないであろう。

実行可能解まで分枝すれば、目的関数値を計算して暫定最適解と比較する。目的関数値の計算には期間の数だけの最短経路探索を行わなければならない。そこで、計算時間を短くするために、その実行可能解の目的関数の下限値を調べ、最適解である可能性のあるものだけについて目的関数値を計算する。この下限値は、各期間で除かれているリンクによる目的関数の増加の下限値を加えて求める。すなわち、式（5.39）、（5.40）である。

$$Z^{tl} = Z^{t0} + w^t \sum_{k=1}^{m} f_k^t (1 - x_k^t) \tag{5.39}$$

$$Z_2^l = \sum_{t=1}^{T} Z^{tl} \tag{5.40}$$

ここに、Z^{tl}: 実行可能解に対する期間tの部分目的関数Z^tの下限値

Z^{t0}: 期間tにおける最大ネットワークの部分目的関数値

Z_2^l : 実行可能解に対する目的関数の下限値

この下限値は必ずしも強力ではない。必要ならば、次の操作を行って下限値を強めればよい。すなわち、実行可能解への分枝の途中で、ある期間に対する分枝が終わったとき、その期間の部分目的関数値を計算しておく。そして、これを式（5.39）のZ^{tl}の代わりに使うのである。その期間におけるネットワークの状態が変化しないなら、次に求めた実行可能解に対してもこの値が使用できる。

厳密解法のアルゴリズムは次のようになる。

ステップ1：各期間において、それぞれのリンクを除くことによる総走行時間の増加の下限値f_k^tを求める。

ステップ2：各期間において、f_k^t / c_kの大きさで並べたときのリンクの順序を求める。

ステップ3：近似解法によって初期解$\boldsymbol{x}^* = \{x_1^*, x_2^*, \cdots, x_m^*\}$を求め、その目的関数値を$Z^*$、最終期間$T$のネットワークに含まれるリンクの建設費の合計を$C^*$とする。

ステップ4：すべての変数を自由変数、すなわち$x_k \leftarrow -1$とする。

ステップ5：その節点における固定変数を制約に加え、補助問題の近似解を求める。

ステップ6：補助問題の近似解にしたがい、実行可能解まで分枝する。

ステップ７：実行可能解に対する目的関数の下限値 Z_2^l を求める。

ステップ８：$Z_2^l \leqq Z^*$ ならばステップ９へ、$Z_2^l > Z^*$ ならばステップ11へ進む。

ステップ９：実行可能解の目的関数値 Z、最終期間 T のネットワークに含まれるリンクの建設費の合計 C を求める。

ステップ10：$Z < Z^*$ のとき、あるいは $Z = Z^*$ で $C < C^*$ のとき、Z^* を Z で、C^* を C で、$\boldsymbol{x}^*$ を $\boldsymbol{x}$ で置き換える。

ステップ11：その節点への分枝変数 x_k の値が１ならばステップ12へ、１でなければステップ13へ進む。

ステップ12：$x_k \leftarrow -1$ として分枝変数を自由変数にし、ひとつ手前の節点へもどる。そして、ステップ11へもどる。もどる節点がないときは計算を終える。

ステップ13：分枝変数に対して $x_k \leftarrow x_k - 1$ とし、他の節点へ分枝する。ステップ５へもどる。

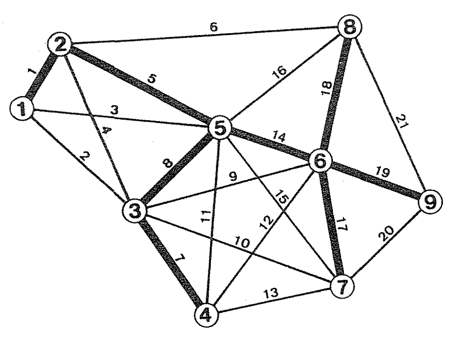
計算を終えたときの $\boldsymbol{x}^*$ が最適解である。

ここで提案した解法は基礎的なものであり、計算時間の短縮については十分でない。探索木のある節点から分枝して得られる実行可能解の目的関数の下限値を用い、分枝を早い段階で打ち切れるようにしなければならない。強力な下限値を求めることが課題である。また近似解法については、容易に求まる近似解の近傍で局所最適化を行う方法などの適用が考えられる。簡単には、式（５.38）で定義した Z_1^l を下限値のように用いることにより、ここで提案した厳密解法を近似解法にすることもできる。

５.４.４　交通網の段階建設問題の計算例

ここでは交通網の段階建設問題の例題を解き、問題５.４の定式化によって妥当な解が得られるかどうかを調べる。

例題のネットワークには例題３.４のものを用いる。これを例題５.２とし、最大ネットワークを**図－５.６**に示す。図に太い実線で表わしたリンクは建設開始時に存在するものとする。リンクの長さと建設費は**表－５.３**のようにする。建設期間は３期間とし、需



図―５.６　例題５.２の最大ネットワーク

表―５.３　例題５.２のリンクの長さと建設費

リンク	端	点	長　さ	建設費
1	1	2	199	
2	1	3	401	40
3	1	5	526	53
4	2	3	483	48
5	2	5	477	
6	2	8	772	77
7	3	4	338	
8	3	5	318	
9	3	6	514	51
10	3	7	590	59
11	4	5	502	50
12	4	6	514	51
13	4	7	373	37
14	5	6	283	
15	5	7	535	54
16	5	8	438	44
17	6	7	340	
18	6	8	363	
19	6	9	313	
20	7	9	325	33
21	8	9	511	51

要交通量は表－５．４～５．６のものを用いる。期間１の需要交通量は発生、集中交通量を乱数で決め、重力モデル式を仮定して求めたものである。期間２、３の需要交通量は発生、集中交通

表—5. 4　例題 5. 2 の期間 1 の需要交通量

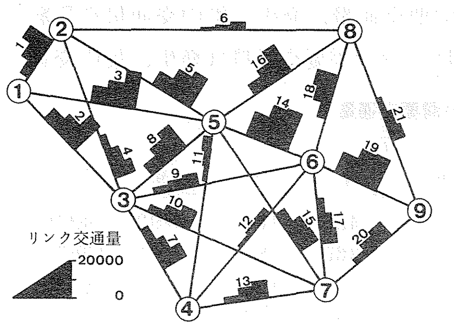
発生	集中 1	2	3	4	5	6	7	8	9	合計
1	4519	2874	1186	758	2604	673	485	1014	93	14206
2	2240	3252	794	532	2068	522	360	831	71	10670
3	2312	1987	2432	1420	4145	1036	771	1325	134	15562
4	470	424	452	629	921	303	297	358	43	3897
5	1042	1062	851	594	3568	786	433	974	84	9394
6	715	712	564	518	2085	1052	503	964	110	7223
7	1516	1443	1236	1494	3384	1482	1862	1561	256	14234
8	808	850	542	461	1940	724	398	1763	99	7585
9	1840	1811	1365	1382	4178	2049	1620	2458	521	17224
合計	15462	14415	9422	7788	24893	8627	6729	11248	1411	99995

表—5. 5　例題 5. 2 の期間 2 の需要交通量

発生	集中 1	2	3	4	5	6	7	8	9	合計
1	6475	6812	2180	1900	4561	927	1114	2586	141	26696
2	2569	6174	1170	1069	2902	576	662	1699	87	16908
3	2839	4037	3832	3053	6225	1223	1518	2899	175	25801
4	5557	830	687	1304	1334	345	563	756	54	6430
5	1072	1807	1123	1070	4486	777	715	1783	92	12925
6	883	1455	894	1120	3149	1249	997	2121	144	12012
7	2090	3292	2187	3605	5707	1965	4117	3835	374	27172
8	614	1069	529	613	1803	529	485	2387	80	8109
9	2275	3704	2165	2991	6317	2436	3212	5413	683	29196
合計	19374	29180	14767	16725	36484	10027	13383	23479	1830	165249

表—5. 6　例題 5. 2 の期間 3 の需要交通量

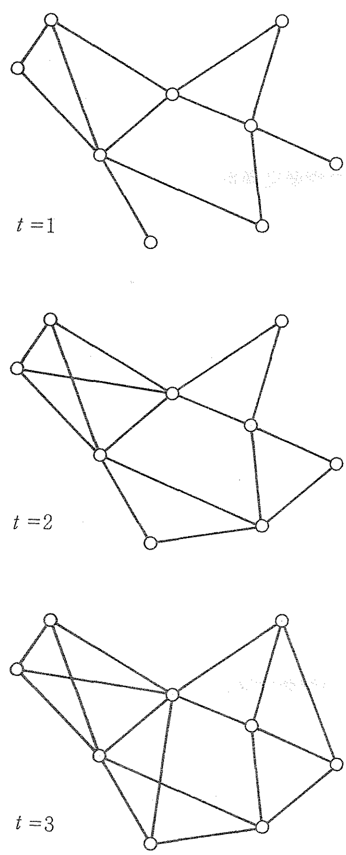
発生	集中 1	2	3	4	5	6	7	8	9	合計
1	9940	11370	5993	4533	7196	1539	2586	5729	285	49171
2	3530	9223	2877	2282	4097	857	1375	3369	156	27766
3	2748	4249	6649	4599	6198	1283	2226	4055	223	32230
4	544	882	1203	1983	1341	365	834	1068	70	8290
5	1669	3060	3132	2589	7179	1309	1684	4008	188	24818
6	888	1592	1613	1753	3259	1361	1519	3082	190	15257
7	1760	3014	3300	4725	4944	1792	5254	4666	414	29869
8	610	1156	941	947	1843	569	730	3428	104	10328
9	1977	3500	3372	4044	5646	2292	4227	6795	781	32634
合計	23666	38046	29080	27455	41703	11367	20435	36200	2411	230363



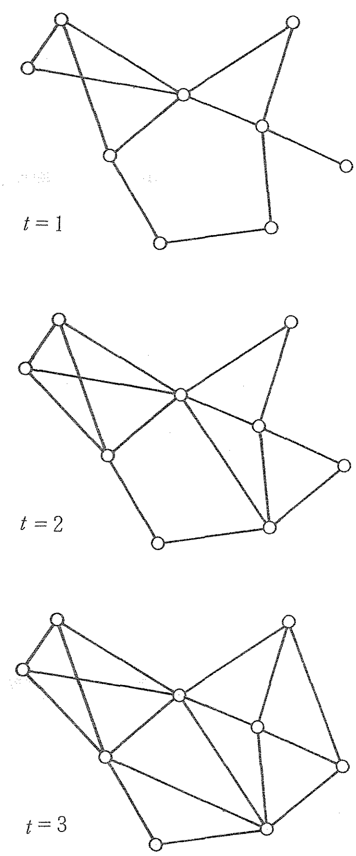
図—5.7　例題5.2の需要交通量の最大ネットワークへの配分結果

量の成長率を乱数で決め、フレータ法で計算したものである。各期間の需要交通量を最大ネットワークに配分すると、図－５．７のようになる。各期間の予算は総建設費の30%、20%、20%にあたる195、130、130とする。

これらのデータを使い、ケース１として最終期間の需要交通量だけを考えて解く。すなわち、各期間の目的関数における重みを０、０、１とする。そして、ケース２として各期間の重みを等しく１、１、１として解く。



図—5.8　例題5.2のケース１の最適解



図—5.9　例題5.2のケース２の最適解

計算の結果、図－５．８、５．９のネットワークがケース１、ケース２の最適解として得られた。これらのネットワークでの各期間の総走行距離は表－５．７のとおりである。この結果は、建設途中の段階でのネットワークに対する需要交通量を考慮すれば最適交通網が異なる例を示している。したがって、交通網を段階的に建設する場合、問題５．４のような定式化が必要であることがわかる。

表―5.7　例題5.2の最適解での総走行距離

期　間	総走行距離（$\times 10^5$） ケース1	ケース2
1	5020	4930
2	7800	7843
3	10472	10475
合　計	23292	23248

5.4.5　むすび

この節では、交通網を段階的に建設する問題の解法を考察した。段階建設問題の特徴は、建設途中の交通網に対する需要交通量を考慮することと、リンクをいつつくるかを決めることである。このため、選択肢はリンクの建設の可否の2種類でなく、それ以上の数になる。したがって、通常の最適交通網構成問題の解法をそのまま使うことはできない。そこで、この問題に適した分枝方法を用いる厳密解法を提案した。そして、簡単な計算例を示した。

ここでは、交通網の段階建設問題への最適交通網構成手法の適用に関する基礎的な考察を行ったにすぎない。実際的な問題を扱うためには近似解法の作成が必要である。そして、建設途中の交通網に対する需要交通量をどのような重みで考慮すればよいかも検討してゆかなければならない。

5.5　需要交通量が不確実な問題への拡張[12)]

5.5.1　はじめに

この節では、不確実な需要交通量を用いて最適な交通網を決める問題を示し、その解法を検討する。

最適交通網構成問題においては、使用するデータの誤差は一般に考慮されていない。しかし、推定値やつねに変動している値を用いる場合、その影響により誤った選択をすることがある。そのような問題の代表的なものに、需要交通量が不確実なときに、その誤差や変動を考えなければならない問題がある。ここでは、この最適交通網構成問題について述べる。そして、解法についての基礎的な考察を行う。

5.5.2では、需要交通量の誤差や変動が最適交通網の決定に及ぼす影響について述べる。そして、不確実な需要交通量を用いる最適交通網構成問題を定式化する。

5.5.3では、需要交通量が不確実な問題の解法を検討する。通常の最適交通網構成問題に対する解法をもとにし、どのように直せばこの問題が解けるかを述べる。

5.5.4では、需要交通量が不確実な問題の計算例を示す。

5.5.2　需要交通量が不確実な場合の最適交通網構成問題

交通網計画を行うときには需要交通量を設定し、それに対して適切な交通網案を作成する方法が用いられる。これは最適交通網構成手法においても同じである。与えられた需要交通量に対して最適な交通網を求めるのである。したがって、解として得られた交通網は前提とした需要交通量に対しては最適であっても、需要交通量が変われば最適である保証はない。交通網の決定に需

要交通量の果たす役割は大きい。

最適交通網構成問題では需要交通量として分布交通量を与える。そして、この分布交通量は推定値であることがほとんどである。推定には誤差が生じるのが普通であり、交通網が完成したときの実際の需要交通量は仮定と大きく異なる可能性もある。さらに分布交通量はつねに一定ではなく、時間的に変動する。時間単位、日単位の変動から季節的な変動まで多様に変化する。このように、需要交通量はひとつの数字で表わしきれず、誤差や変動を含むと考えなければならない。

通常の最適交通網構成問題では、与えられた需要交通量は確定的なものとしている。しかし、需要交通量が推定誤差や時間的変動を含む不確実なものであるならば、その不確実さを考慮して交通網を決めなければならないかもしれない。そこで、需要交通量として用いる分布交通量が確率分布をすると仮定し、これが最適交通網の決定に及ぼす影響を調べる。

需要交通量と交通網との間には密接な関係があり、交通網が変われば需要交通量も変化すると考られている。したがって、どのような交通網になるかが決まっていない時点においては、需要交通量が不確実であることになる。だが、ここでは交通網の変化による需要交通量の変化は考慮しない。すなわち、この意味での需要交通量の不確実性は扱わないことにする。

分布交通量の確率分布については、いろいろな場合が想定される。大きく分ければ、個々の分布交通量がたがいに独立に分布する場合と、独立でない場合が考えられる。独立でない場合は、いくつかのOD表がそれぞれの生起確率で生じると考えてもよい。しかし、この違いは問題の定式化と解法の作成において決定的なものではない。ここでは、個々の分布交通量が独立に確率分布するとして問題を考える。

需要交通量の不確実さが交通網の決定に及ぼす影響を調べるために、予算制約下で総走行距離を最小化する問題を考える。最適交通網構成問題として定式化すれば、次のようになる。

問題５．５

$$\min \quad Z=\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}\,d_{ij}(\boldsymbol{x}) \tag{5.41}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} c_k\, x_k \leqq C^c \tag{5.42}$$

$$x_k = 0 \ or\ 1 \quad (k=1,\ 2,\cdots,m) \tag{5.43}$$

ここに、Z：総走行距離

q_{ij}：ノード ij 間のトリップ数

d_{ij}：ノード ij 間の最短距離

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_k, \cdots, x_m\}$

：ネットワークの状態を表わし、リンク k がネットワークに含まれるとき $x_k=1$、含まれないとき $x_k=0$ とする。

c_k：リンク k の建設費

C^c：予算すなわちネットワークの建設費の上限値

n：ノード数

m：リンク数

分布交通量 q_{ij} が確率分布をするならば、式（５.41）の目的関数を直さなくてはならない。

このときの一般的な考え方は、総走行距離の期待値を最小化するものである。そうすると、目的関数は式（5.44）になる。ただし、$E\{\ \}$ は $\{\ \}$ の期待値を表わす。そして、q_{ij}の平均値を q_{ij}^{m} とすれば、式（5.44）は式（5.45）になる。

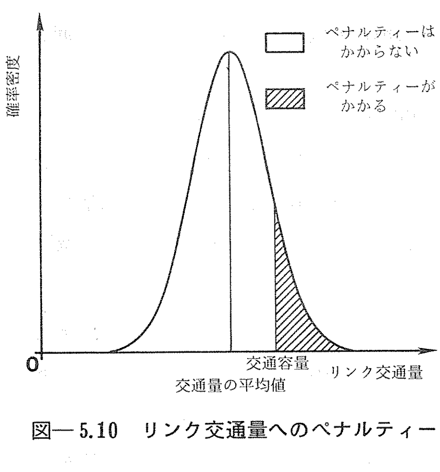
$$\min \quad Z=E\{\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}\, d_{ij}(\boldsymbol{x})\} \tag{5.44}$$

$$\min \quad Z=\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}^{m}\, d_{ij}(\boldsymbol{x}) \tag{5.45}$$

式（5.45）は式（5.41）と同じ形をしており、分布交通量が確率分布をするときも、その平均値を用いて通常の最適交通網構成問題として扱えることを示している。すなわち、総走行距離の期待値の最小化を目的関数とするときは、分布交通量の平均値に対して最適な交通網を求めればよい。

しかしながら、分布交通量の平均値を使って解くだけでよい問題ばかりではない。その例に、リンクの容量制約を伴う問題がある。交通網計画では、リンクの交通量に対する容量の制約をなんらかの形で考慮しなければならないことが多い。たとえば、容量を超える交通量は許されないとする場合や、所要時間の増加となる場合、追加の投資を必要とする場合などがある。明確な制約条件を設けないときは、容量を超える交通量にペナルティーが課せられるのである。そして、容量を超える交通量を許さないとする制約条件も、ペナルティーを課す方法で扱える。そこで、リンクの交通量が容量を超えればペナルティーが課せられる問題を考える。

リンク交通量すなわち配分交通量は分布交通量の和である。したがって、分布交通量が確率分布するときは、リンク交通量も確率分布する。分布交通量の平均値を使って問題を解くときは、リンク交通量も平均値だけが与えられることになる。いま、リンクの交通量と容量との関係が図－5.10のようになったとする。平均値だけで考えれば交通量は容量以下であり、ペナルティーは課せられない。しかし、リンク交通量が容量を超える確率は0でなく、ペナルティーが課せられることが起こり得る。ペナルティーの期待値は0ではないのである。そのため、このような場合に分布交通量の平均値だけで解を求めるのは誤った方法であるといえる。分布交通量の不確実さを考慮しなければ、交通網の正しい評価ができないのである。



図－5.10　リンク交通量へのペナルティー

ここでの目的は、需要交通量の不確実さを考えなければならない最適交通網構成問題を取り上げ、基礎的な考察を行うことである。そこで、リンク交通量が容量を超えたときに、その超過量にペナルティーを課し、総走行距離とペナルティーの和を最小化する問題を考える。この問題は、需要交通量の不確実さを考慮することが必要な、もっとも単純な問題である。これを定式化すると、次のようになる。

問題5.6

$$\min \quad Z=\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}^{m}\, d_{ij}+\sum_{k=1}^{m}\sum_{h=1}^{2} E\{p_{kh}\} \tag{5.46}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} c_k x_k \leqq C^c \tag{5.47}$$

$$x_k = 0 \ \ or\ \ 1 \quad (k=1,2,\cdots,m) \tag{5.48}$$

ここに、Z：総走行距離とペナルティーの和の期待値

q_{ij}^{m}：ノード ij 間のトリップ数の平均値

p_{kh}：リンク k の方向 h の交通量に対するペナルティー

ただし、ペナルティーは次の式（5.49）のとおりとする。

$$p_{kh} = \begin{cases} p_k(v_{kh} - u_k) & if \quad v_{kh} > u_k \\ 0 & if \quad v_{kh} \leqq u_k \end{cases} \tag{5.49}$$

ここに、p_k：リンク k の容量を超える交通量に課せられる単位ペナルティー

v_{kh}：リンク k の方向 h の交通量

u_k：リンク k の交通容量

トリップは最短経路に配分する。d_{ij} 、v_{kh}、p_{kh} はそれぞれネットワークによって決まることになる。目的関数の第１項の総走行距離は問題５．５と同じであり、第２項のペナルティーを含むことが問題５．６の特徴である。ペナルティーの具体的な解釈はしないが、第１項とあわせて交通量と所要時間の関係を表わすと考えてもよいし、必要な追加費用と考えてもよい。また、ここでは交通量の超過量に比例してペナルティーが増加するが、必ずしも比例関係でなくてよい。

分布交通量は、それぞれが独立に確率分布するものとする。そして、計算を簡単にするために、分布交通量の確率分布は正規分布で表わせるとする。

５．５．３　需要交通量が不確実な問題の解法

ここでは、**５．５．２**で示した需要交通量が不確実な最適交通網構成問題を解くために、通常の問題に対する解法をどのように直せばよいかについて述べる。

問題５．６が通常の最適交通網構成問題と異なるのは、目的関数にリンクの交通容量を超えた交通量に対するペナルティーの項を含み、分布交通量が確率分布をすることである。リンクの組み合わせであるネットワークを決めることはまったく同じなので、解法には分枝後退法が適用できる。そこで、総走行距離の最小化を目的関数とする問題の解法をもとに、需要交通量が不確実な問題の解法をつくる。

予算制約下で総走行距離を最小化する問題は問題５．５として示した。この問題の解は最適交通網構成問題の基本問題と同じ考え方で求められる。目的関数の下限値を用い、分枝と実行可能性の検査、最適性の検査を繰り返すのである。問題５．６の目的関数は総走行距離とペナルティーの和で、ペナルティーは非負である。そのため、総走行距離の下限値は目的関数全体の下限値でもある。したがって、実行可能解の目的関数値を計算する部分だけを変えれば、総走行距離の最小化を行う場合と同じ解法が使える。よって、解法の基本的な構成は総走行距離を最小化する問題の解法と同じとし、問題に合わせる変更を加える。

変更を要する部分には目的関数値の計算がある。目的関数値を求めるには、第２項のペナルティーを計算しなければならない。ペナルティーの値を決めるにはリンク交通量が必要なので、最短経路配分を行う。このとき、リンク交通量の平均値だけでなく、どのような分布をするかも求め

る。そして、それをもとにペナルティーの期待値を計算する。目的関数値の計算は繰り返し行うので、ペナルティーの期待値は簡単な計算で求まるようにしなければならない。そこで、次の方法を用いる。

対象としている問題では分布交通量を正規分布で表わせるとした。そのため、分布交通量の和であるリンク交通量の分布も正規分布になる。その平均値と分散は、それぞれ式（5.50）、（5.51）で求まる。

$$v_{kh}^{m}=\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_{ijkh}\, q_{ij}^{m} \qquad (5.50)$$

$$v_{kh}^{v}=\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_{ijkh}\, q_{ij}^{v} \qquad (5.51)$$

ここに、 v_{kh}^{m}：リンク k の方向 h の交通量の平均値

v_{kh}^{v}：リンク k の方向 h の交通量の分散

δ_{ijkh}：ノード ij 間の最短経路にリンク k の方向 h が含まれるとき1、含まれないとき0とする。

リンク k の方向 h の交通量に対するペナルティーの期待値を求める式を導くが、簡単のために v_{kh} を v 、 v_{kh}^{m} を ρ 、 v_{kh}^{v} を σ^2、u_k を u 、p_k を p 、そして $E\{p_{kh}\}$ を P と表わす。リンク交通量の確率分布は式（5.52）の確率密度関数で表わされる。

$$f(v)=\frac{1}{\sqrt{2\pi}\ \sigma}exp\left\{-\frac{(v-\rho)^2}{2\sigma^2}\right\} \qquad (5.52)$$

ペナルティーの期待値は式（5.53）で求まる。そして、規準正規分布を使うために、式（5.54）の変換を行うと、式（5.55）になる。

$$P=\int_{u}^{\infty}p(v-u)f(v)\,dv \qquad (5.53)$$

$$t=\frac{v-\rho}{\sigma},\quad w=\frac{u-\rho}{\sigma} \qquad (5.54)$$

$$P=p\sigma\int_{w}^{\infty}\frac{t}{\sqrt{2\pi}}\,exp\left(-\frac{t^2}{2}\right)dt+p(\rho-u)\int_{w}^{\infty}\frac{1}{\sqrt{2\pi}}\,exp\left(-\frac{t^2}{2}\right)dt \qquad (5.55)$$

ここで、規準正規分布に対して、式（5.56）と式（5.57）で表わされるふたつの関数を定義する。そうすれば、式（5.55）は式（5.58）になる。

$$F_1(x)=\int_{x}^{\infty}\frac{t}{\sqrt{2\pi}}exp\left(-\frac{t^2}{2}\right)dt \qquad (5.56)$$

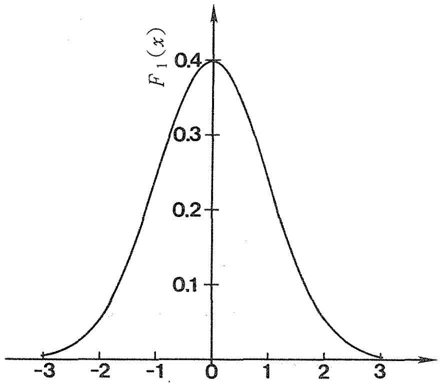
$$F_2(x)=\int_{x}^{\infty}\frac{1}{\sqrt{2\pi}}\,exp\left(-\frac{t^2}{2}\right)dt \qquad (5.57)$$

$$P=p\sigma\,F_1\left(\frac{u-\rho}{\sigma}\right)+p(\rho-u)\,F_2\left(\frac{u-\rho}{\sigma}\right) \qquad (5.58)$$

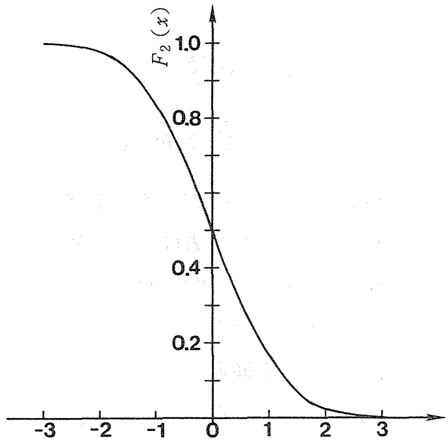
記号を元にもどすと、ペナルティーの期待値を求める式は式（5.59）になる。

$$E\{p_{kh}\}=p_k\sqrt{v_{kh}^{v}}\,F_1\left(\frac{u_k-v_{kh}^{m}}{\sqrt{v_{kh}^{v}}}\right)+p_k(v_{kh}^{m}-u_k)\,F_2\left(\frac{u_k-v_{kh}^{m}}{\sqrt{v_{kh}^{v}}}\right) \qquad (5.59)$$

$F_1(x)$ は x の超過量の期待値を表わし、図－5.11に示すものである。$F_2(x)$は x の超過確率を表わし、図－5.12に示すものである。ペナルティーの期待値の計算を簡単にするため、いくつかの x の値に対する $F_1(x)$, $F_2(x)$ の値をあらかじめ用意しておき、線形補間で関数値が求まるようにしておく。いわゆるテーブル関数にしておくのである。



図―5.11　関数 $F_1(x)$



図―5.12　関数 $F_2(x)$

解法で変更を要するもうひとつの部分は目的関数の下限値である。リンク交通量が容量を超えることが少ない場合はよいが、目的関数値に占めるペナルティーの値の割合が大きくなれば、ペナルティーの下限値を考えなくてはならない。しかし、交通配分を行わずにリンク交通量を予測することは困難である。そして、ネットワークからリンクを除くことによって、つねにペナルティーが増加するともいえない。そのため、総走行距離のように容易にペナルティーの下限値を求めることはできない。そこで、強力な下限値が求まらなくてもよいから簡単な方法を使うことにする。すなわち、個々のリンクのペナルティーをもとにするのでなく、カットを使用して下限値を求める方法である。

あるカットにより、ネットワークに属するノードは2個の集合に分けられる。起終点のノードが別のノード集合に属するとき、そのトリップの経路は必ずカットを横切る。そこで、式（5.60）でカットを構成するリンクの交通量の和の下限値が求まる。一方、あるネットワークの状態において、カットに属するリンクの交通容量の和は式（5.61）で計算できる。

$$V_{rh} = \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ijrh}\, q_{ij} \qquad (5.60)$$

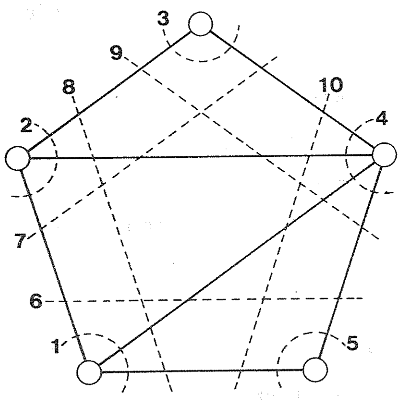
$$U_r = \sum_{k=1}^{m} \zeta_{kr}\, x_k\, u_k \qquad (5.61)$$

ここに、 V_{rh}：カット r の方向 h の交通量の下限値

U_r：カット r の交通容量

γ_{ijrh}：ノード ij 間の経路が必ずカット r を方向 h に横切るとき1、そうでないとき0とする。

ζ_{kr}：リンク k がカット r に含まれるとき1、含まれないとき0とする。



図―5.13　カットの例

カットを用いてペナルティーの下限値を求めるひとつの方法として、同一のリンクが複数のカットに属することのないカットの集合をつくる。たとえば、**図－5.13**の例では｛1、3｝｛2、5｝｛3、5｝｛3、6｝などがこれに相当する。そうすると、あるカット集合の中では1本のリンクのペナルティーが複数のカットで加算されることがない。そのため、ペナルティーである目的関数の第2項の下限値は、式（5.62）～（5.65）を使って求めることができる。

$$Z^{l}_{2s} = \sum_{r \in A_s} \sum_{h=1}^{2} E\{P_{rh}\} \qquad (5.62)$$

$$P_r = min\{p_k | \zeta_{kr} = 1,\ x_k = 1\} \qquad (5.63)$$

$$P_{rh} = \begin{cases} P_r(V_{rh} - U_r) & if\ \ V_{rh} > U_r \\ 0 & if\ \ V_{rh} \leqq U_r \end{cases} \qquad (5.64)$$

$$Z^{l}_{2} = max(Z^{l}_{2s}) \qquad (5.65)$$

ここに、　Z^{l}_{2s}：カット集合 s で求めた目的関数の第2項の下限値

Z^{l}_{2}：目的関数の第2項の下限値

A_s：カット集合 s

P_{rh}：カット r の方向 h に含まれるリンクの交通量に対するペナルティーの下限値

P_r：カット r に含まれるリンクの単位ペナルティーの最小値

期待値の計算は、各リンクのペナルティーの期待値の計算と同じ方法を用いればよい。

ネットワークからリンクを除いても、ペナルティーが必ず増加するとは限らないことを述べた。このため、分枝の打ち切りの判定にも変更を要する。総走行距離だけが目的関数のときは、ある実行可能解からさらにリンクを除いた解を調べる必要はなかった。それは、リンクを除けば必ず目的関数値が大きくなるからである。しかし、ペナルティーを目的関数に含むときは、リンクを除いた方が目的関数値が小さいことが起こり得る。そこで、単に実行可能解に達したという理由では分枝を打ち切れない。さらに分枝を続けなければならない。この場合の分枝の打ち切りの判定は、式（5.66）あるいは式（5.67）が満たされるかどうかで決める。

$$\sum_{i=1}^{n} \sum_{j=1}^{n} q^{m}_{ij}\, d_{ij} + Z^{l}_{2} > Z^{*} \qquad (5.66)$$

$$Z^{l}_{1} + Z^{l}_{2} > Z^{*} \qquad (5.67)$$

ここに、Z^{*}：計算途中の段階における暫定最適解の目的関数値

Z^{l}_{1}：目的関数の第1項の下限値

Z^{l}_{1} は Hoang の方法[11]に準じて求めるものとする。これらの式を満たす場合は探索木上のひとつ手前の節点へもどり、もう一方の節点へ分枝する。

基本的な計算手順は連結網化分枝と実行可能化分枝を行うアルゴリズムに従うとすると、需要交通量が不確実な問題を解くアルゴリズムは次のようになる。

ステップ1：各リンクを除くことによる総走行距離の増加の下限値 f_k を求める。

ステップ2：$f_1/c_1 \leqq f_2/c_2 \leqq \cdots \leqq f_m/c_m$ となるようにリンクを並べかえる。

ステップ3：カットの交通量の下限値の平均値 V_{rh}^m と分散 V_{rh}^v を求める。

ステップ4：近似解法によって初期解 $\boldsymbol{x}^*=\{x_1^*, x_2^*, \cdots, x_m^*\}$ を求め、その目的関数値を Z^*、ネットワーク建設費を C^* とする。

ステップ5：すべての変数を自由変数、すなわち $x_k \leftarrow -1$ とする。

ステップ6：節点の表わす解集合の中での最小建設費の連結網を求め、そのネットワーク建設費を C^s とする。最小建設費の連結網が求まればステップ7へ、連結網がなければステップ9へ進む。

ステップ7：$C^s \leqq C^c$ ならばステップ8へ、$C^s > C^c$ ならばステップ9へ進む。

ステップ8：最小建設費の連結網を構成するリンクで $x_k = -1$ のものを建設費の小さい順に分枝変数として選び、$x_k \leftarrow 1$ として分枝する。ステップ13へ進む。

ステップ9：その節点への分枝変数 x_k の値が1ならばステップ10へ、0ならばステップ11へ進む。

ステップ10：ひとつ前の節点へあともどりして $x_k \leftarrow 0$ とし、もう一方の節点へ分枝する。ステップ6へもどる。

ステップ11：$x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。

ステップ12：その節点が探索を開始した節点ならば計算を終える。そうでなければステップ9へもどる。

ステップ13：$x_k = 0$ あるいは $x_k = 1$ の固定変数を制約に加え、総走行距離の下限値を求めるための補助問題を解く。最適解が求まればステップ14へ、実行可能解がなければステップ23へ進む。

ステップ14：総走行距離の下限値 Z_1^l を計算する。

ステップ15：$Z_1^l \leqq Z^*$ ならばステップ16へ、$Z_1^l > Z^*$ ならばステップ23へ進む。

ステップ16：自由変数からリンク番号の小さい順に分枝変数 x_k を選び、ネットワーク建設費が制約値以下になるまで $x_k \leftarrow 0$ として分枝する。

ステップ17：ペナルティーの下限値 Z_2^l を求める。

ステップ18：$Z_1^l + Z_2^l \leqq Z^*$ ならばステップ19へ、$Z_1^l + Z_2^l > Z^*$ ならばステップ23へ進む。

ステップ19：$x_k \neq 0$ であるリンクからなるネットワークの目的関数値 Z とネットワーク建設費 C を求める。

ステップ20：(Z, C) が辞書編集的に (Z^*, C^*) より小さければ、Z^* を Z で、C^* を C で、$\boldsymbol{x}^*$ を $\boldsymbol{x}=\{|x_k|\}$ で置き換える。ステップ21へ進む。

ステップ21：総走行距離を Z_1 とし、$Z_1 + Z_2^l \leqq Z^*$ ならばステップ22へ、$Z_1 + Z_2^l > Z^*$ ならばステップ23へ進む。

ステップ22：自由変数のなかで、リンク番号が最小のものを分枝変数に選び、$x_k \leftarrow 0$ として

分枝する。そして、ステップ19へもどる。もし自由変数がないときは、ステップ23へ進む。

ステップ23：その節点への分枝が連結網化分枝によるものならばステップ９へもどる。そうでなければステップ24へ進む。

ステップ24：その節点への分枝変数 x_k の値が０ならばステップ25へ、１ならばステップ26へ進む。

ステップ25：ひとつ前の節点へあともどりして $x_k \leftarrow 1$ とし、もう一方の節点へ分枝する。分枝が実行可能化分枝によるものであったならステップ13へもどる。そうでないときはステップ22へもどる。

ステップ26：$x_k \leftarrow -1$ とし、ひとつ前の節点へあともどりする。ステップ23へもどる。

計算を終えたときの $\boldsymbol{x}^*$ が最適解である。初期解を求める近似解法には backward 法を使用する。これは目的関数値の計算の部分が異なるが、基本問題に対するものと同じ計算手順である。

５．５．４　需要交通量が不確実な問題の計算例

ここでは問題５．６の例題を解き、需要交通量の不確実さを考慮するかどうかで、異なる交通網案が得られる場合があることを示す。

例題のネットワークには例題３．４のものを用いる。これは、**５．４．４**でも例題５．２として用いたものである。リンクの建設費と交通容量、単位ペナルティーなどは**表－５．８**のように定め、これを例題５．３とする。需要交通量は例題５．２で使ったものを平均値として採用する。

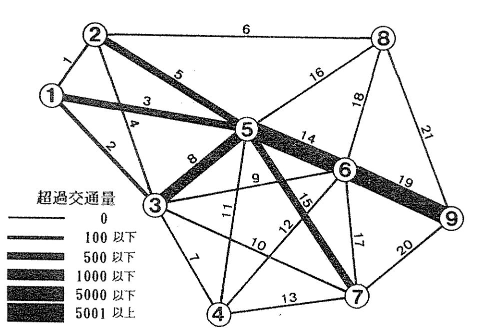
表—5.8　例題5.3のリンクのデータ

リンク	端	点	長さ	建設費	交通容量	ペナルティー
1	1	2	199	199	5000	100
2	1	3	401	401	5000	201
3	1	5	526	526	5000	263
4	2	3	483	483	5000	242
5	2	5	477	477	5000	239
6	2	8	772	772	5000	386
7	3	4	338	338	5000	169
8	3	5	318	318	5000	159
9	3	6	514	514	5000	257
10	3	7	590	590	5000	295
11	4	5	502	502	5000	251
12	4	6	514	514	5000	257
13	4	7	373	373	5000	187
14	5	6	283	283	5000	142
15	5	7	535	535	5000	268
16	5	8	438	438	5000	219
17	6	7	340	340	5000	170
18	6	8	363	363	5000	182
19	6	9	313	313	5000	157
20	7	9	325	325	5000	163
21	8	9	511	511	5000	256

表―5.9　例題5.3の需要交通量

発生	集					中				
	1	2	3	4	5	6	7	8	9	合　計
1	4519	2874	1186	758	2604	673	485	1014	93	14206
	4315	2792	1172	752	2536	669	483	1004	93	
2	2240	3252	794	532	2068	522	360	831	71	10670
	2190	3146	788	529	2025	519	359	824	71	
3	2312	1987	2432	1420	4145	1036	771	1325	134	15562
	2259	1948	2373	1400	3973	1025	765	1308	134	
4	470	424	452	629	921	303	297	358	43	3897
	468	422	450	625	913	302	296	357	43	
5	1042	1062	851	594	3568	786	433	974	84	9394
	1031	1051	844	591	3441	780	431	965	84	
6	715	712	564	518	2085	1052	503	964	110	7223
	710	707	561	515	2042	1041	501	955	110	
7	1516	1443	1236	1494	3384	1482	1862	1561	256	14234
	1493	1422	1221	1472	3270	1460	1827	1537	255	
8	808	850	542	461	1940	724	398	1763	99	7585
	802	843	539	459	1902	719	396	1732	99	
9	1840	1811	1365	1382	4178	2049	1620	2458	521	17224
	1806	1778	1346	1363	4004	2007	1594	2398	518	
合　計	15462	14415	9422	7788	24893	8627	6729	11248	1411	99995

上段：平均　　下段：分散（× 100）

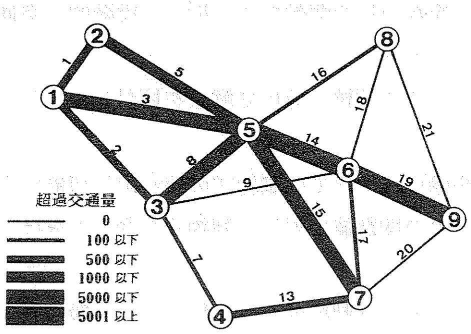


図―5.14　例題5.3の最大ネットワークへの配分結果

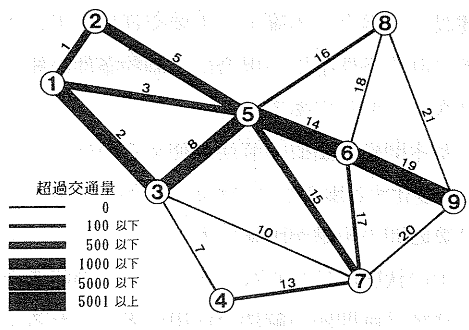
分散は多項分布に従う抽出を行うと仮定し、抽出率1%の標本からOD表を作成する場合を想定して表－5.9のようにする。この需要交通量を最大ネットワークに配分すると、交通容量を超えるリンク交通量の期待値は図－5.14のようになる。予算は最小限必要な建設費を0%、最大ネットワークの建設費を100%にしたとき、60%にあたる6485とする。

まずケース1として需要交通量の不確実さを考慮しない場合の解を求める。このときは需要交通量の分散を0とし、平均値だけを用いる。つぎに、ケース2として需要交通量の不確実さを考慮して解を求める。

計算の結果、それぞれ図－5.15、5.16のネットワークが最適解として得られた。これらの図には、各リンクの超過交通量の期待値を示してある。ケース1の場合は、需要交通量の分散も



図― 5.15　例題 5. 3 のケース 1 の最適解



図― 5.16　例題 5. 3 のケース 2 の最適解

表― 5.10　例題 5. 3 の最適解の目的関数値
(単位：10^3)

目的関数の項	ケース 1	ケース 2
総走行距離	45968	45855
ペナルティー	3202	3276
合　　計	49170	49131

使って求め直したものである。これから、それぞれのネットワークの目的関数値を計算し直すと、表－５.10のようになる。これは、分散を０として解を求めるとペナルティーを過小に見積り、最適解でない解が得られる場合があることを示している。需要交通量の不確実さを考慮しないで交通網を決めると、損失を被る可能性があることがわかる。

５.５.５　むすび

この節では需要交通量の不確実さを考慮しなければならない最適交通網構成問題を示し、その解法を考察した。需要交通量の不確実さによって交通網の決定が影響される代表的な例として、リンクの交通量に対して容量制約が課される問題を取り上げた。すなわち、予算制約を制約条件とし、総走行距離とリンクの交通量の超過に対するペナルティーの和の期待値の最小化を目的関数とする問題である。そして、この問題を対象とする解法を提案した。計算例では需要交通量の不確実さを考慮しなければ損失を被る可能性があることを示した。

この問題の解法の特徴はペナルティーの計算にある。最適交通網構成問題を解くときには、目的関数値の計算を頻繁に行う。そのため、期待値の計算にテーブル関数を使うなどして計算時間の短縮を図った。そして、ペナルティーの下限値を求めるのに総走行距離の下限値の場合のような方法が使えないため、カットを用いる方法を採用した。

ここでの考察の結果、交通網計画を行うときには需要交通量の精度や変動量を調べ、それが交通網の決定に及ぼす影響を検討する必要があることが明らかになった。そして、場合によっては、ここで示したような解法を使わなければならない。その際は、ここでの考察をもとに近似解法を作成しなければならないであろう。

５.６　結　　語

この章では、最適交通網構成問題を基本問題から拡張したときに、どのように解法を修正しなければならないかを考察した。取り上げたのは、現実的な問題を扱うために必要となる基本的な

場合である。すなわち、複数の制約条件を持つ場合、複数の目的関数を持つ場合、段階的に交通網を建設する場合、不確実な需要交通量を用いる場合である。

複数の制約条件を持つ場合は、制約条件の組み合わせで4種類に分けて解法を検討した。その結果は次のとおりである。

① 基本問題と類似の解法が使えるのは、リンクを除けばすべての制約式の値が実行可能な方向へ変化する場合だけである。しかし、ネットワークの関数値に対する制約式を含むときは、計算時間の短縮が困難である。

② ①の状態にはならないが、リンクの組み合わせに関する制約式だけの場合は、一般の0－1整数計画問題の解法の応用が考えられる。

③ これら以外の場合には、完全列挙法に近い解法を用いなくてはならない。

そして、基本問題と類似の解法が使える問題を対象に解法を検討した。この問題は、制約式が多次元ナップザック問題と同じ形になるものである。これに対し、多次元ナップザック問題の近似解法を用いて実行可能化分枝を行う厳密解法と、それを手直しした近似解法を提案した。

複数の目的関数を持つ多目的最適交通網構成問題の解法に関しては、次のことを得た。

① 目的関数のスカラ化手法は最適交通網構成問題に適用できる。この場合、繰り返し計算をする必要性が大きいことへの対策を考えなければならない。

② 一般の多目的計画問題に対する対話型最適化手法の適用は困難である。基本的な考え方をもとに、最適交通網構成問題に適した手法を開発しなければならない。

③ 選好最適化自体にも近似解法を用いることを考えてよい。

これらの点をもとに、多目的最適交通網構成問題の解を得るための計算手順を一般的な形で示した。これは、ある近似解の近傍に解の探索範囲を限定して実行可能解を列挙し、そこから非劣解を求めておき、目的関数のスカラ化手法を適用するものである。複数の目的関数を持つ問題を実用的に扱うには、このような方法が必要なのである。

交通網を段階的に建設する問題については、建設期間全体を考えて、それぞれの期間のネットワークを決める問題を定式化した。この問題は、リンクをつくるかどうかだけでなく、どの期間につくるかを決めることが特徴である。そのために解の探索木が大きく異なるが、これに適する分枝方法を提案した。

需要交通量が不確実な問題では、その不確実さがネットワークの決定に及ぼす影響について調べた。そして総走行距離を最小化するようなときにはよいが、リンク交通量に容量制約が課せられる場合には需要交通量の不確実さを考慮する必要があることを述べた。そのような問題のなかで簡単なものとして、リンク交通量が容量を超えたときにペナルティーを課し、総走行距離とペナルティーの和の期待値を最小化する問題を考えた。これに対する解法では、ペナルティーの期待値を計算する簡単な方法と、カットを設けてペナルティーの下限値を求める方法を提案した。さらに、需要交通量の不確実さを考慮しないで交通網を決めると、損失を被る可能性があることを計算例で示した。

現実の交通網計画に最適交通網構成手法を適用するときは、ここで示した拡張を組み合わせなければならないことがあると考えられる。その場合、それぞれの拡張に対する解法を組み合わせ

るだけではすまず、解を得ることの困難さが急激に増大するかもしれない。最適交通網構成手法の基本的な考え方と、問題を拡張したときの対処方法の基礎を心得たうえで、個々の交通網計画問題に適した解法を探すことが必要である。

参考文献

1) 枝村俊郎・森津秀夫・土井元治：最適ネットワーク問題の列車運行パターン最適化への応用，建設工学研究所報告，第20号，pp.81～90，昭和53年11月.

2) Senju、S. and Y. Toyoda : An approach to linear programming with 0－1 variables, Management Science, Vol. 15, №4, pp.B－196～B－207, December, 1968.

3) Toyoda, Y. : A simplified algorithm for obtaining approximate solutions to zero－one programming problems, Management Science, Vol. 21, №12, pp.1417～1427, August, 1975.

4) 大根田洋祐・渡辺　隆・森地　茂：環境要因を考慮した街路網計画手法について，土木学会第31回年次学術講演会講演概要集，第4部，pp.148，昭和51年10月.

5) 日野泰雄・西村　昂：複数の目標を考慮した最適ネットワーク構成に関する一考察，土木学会第32回年次学術講演会講演概要集，第4部，pp.23～24，昭和52年10月.

6) 吉川和広・小林潔司・川合紀章：目標計画法による広域的な幹線道路網整備計画問題に関するシステム分析，土木学会第34回年次学術講演会講演概要集，第4部，pp.49～50，昭和54年10月.

7) 伏見多美雄・山口俊和：複数の目標をバランスよく達成させるための数理計画的方法，経営科学，第19巻，第2号，pp.88～102，1975年4月.

8) 志水清孝：多目的と競争の理論，共立出版，昭和57年6月.

9) 杜若善彦・森津秀夫：複数目標下の最適道路網計画モデル，土木学会第37回年次学術講演会講演概要集，第4部，pp.397～398，昭和57年10月.

10) 枝村俊郎・森津秀夫・亀山寿仁：交通ネットワークの最適段階建設，建設工学研究所報告，第22号，pp.255～265，昭和55年12月.

11) Hoang, H. H. : A computational approach to the selection of an optimal network, Management Science, Vol. 15, №5, pp.488～498, January, 1973.

12) 枝村俊郎・森津秀夫・冨村彰廣：不確実性下における最適交通ネットワークの採択について，土木学会第32回年次学術講演会講演概要集，第4部，pp.56～57，昭和52年10月.

第６章　最適交通網構成手法の応用

６．１　概　　説

この章では、いろいろな交通網計画問題への最適交通網構成手法の応用について述べる。第５章までにおいて基本的な問題を対象に解法を改良し、その拡張を検討した。しかし、現実の交通網計画へ適用する際は、この手法に適するように問題を定式化しなければならない。そして、解法も問題にあわせて手直ししなければならない。ここではそのようにして現実的な交通網計画問題を解き、最適交通網構成手法の実用性を示す。さらに、これまで交通網計画問題として扱われていなかった問題へも適用し、応用の可能性が大きいことを示す。

６．２では、鉄道、新交通システム、バスの３種類の大量輸送機関を適切に組み合わせる地域交通網計画問題への応用について述べる。

６．３では、騒音と排出ガスの影響をも考慮して道路網を計画する問題へ適用する。この問題は多目的最適交通網構成問題として解く。そして、交通配分に等時間配分を用いることが特徴である。

６．４では、道路網の災害に対する信頼性を最適化する問題への適用について述べる。これは、災害時においても道路網がその機能を果たせるように、道路網の形状とリンクの災害に対する強さを計画するものである。

６．５では、配分対象道路網を最適交通網構成手法を用いて作成する方法を示す。配分対象道路網はどのようなものであればよいかを検討し、客観的な基準で道路網案を作成する手法を提案する。

６．６では、バス路線網計画問題への適用を行う。バス路線網計画では、バス道路網、路線系統、運行回数を決める。このうち、バス道路網と路線網の決定に最適交通網構成手法を使用する。とくに、路線網を決める問題は路線系統の最適な組み合わせを求めようとするものであり、リンクの組み合わせを決めるこれまでの最適交通網構成問題と異なる。

６．７では、この章で得られた成果をまとめる。

６．２　地域交通網計画問題への適用 [1)]

６．２．１　はじめに

この節では、大量輸送機関に対する需要交通量が与えられたとき、適切な交通機関を組み合わせて地域の交通網を計画する問題を考える。

大量輸送機関には鉄道、バス、モノレール、あるいは新交通システムなどがある。これらの交通機関をどのように設けるかは、地域全体として最適な交通網になるように決めなければならない。そこで、複数の交通機関のネットワークを個別に決めるのでなく、対象とする地域の交通網を一度に決定する問題をつくる。ここでは鉄道、新交通システム、バスを組み合わせる問題を定式化し、その解法を示す。そして、ケーススタディを行う。

６．２．２では、ここで対象とする地域交通網計画問題について述べ、これを解くのに必要な、基本問題に対する近似解法の修正を行う。

6.2.3では、ケーススタディとして北摂・北神地域の交通網計画を取り上げ、最適交通網構成手法を適用する。

6.2.2　地域交通網計画問題とその解法

ここでは、対象とする地域交通網計画問題について述べる。そして、最適交通網構成手法を用いるとき、基本問題に対する解法をどのように修正すればよいか示す。

ここで考えるのは、地域における需要交通量が与えられたとき、適切な大量輸送機関網を計画することである。大量輸送機関は鉄道、新交通システム、バスの3種類とし、計画対象地域にこれらの路線を表わすリンクの組み合わせである交通網を設定する。ネットワーク建設費の予算制約の下で、総所要時間を最小化するものとすれば、問題は次のように定式化できる。

問題6.1

$$\min \quad Z=\sum_i \sum_j q_{ij}\, t_{ij}(\boldsymbol{x}) \tag{6.1}$$

$$\text{s.t.} \quad \sum_k c_k x_k \leqq C \tag{6.2}$$

$$x_k = 0 \ \mathit{or}\ 1 \tag{6.3}$$

ここに、Z：総所要時間

q_{ij}：ノードij間のトリップ数

t_{ij}：ノードij間の最短所要時間

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_k, \cdots\}$

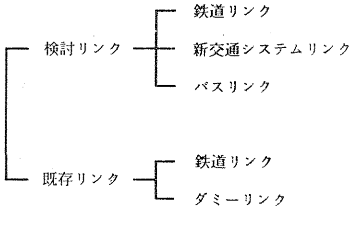
：ネットワークの状態を表わし、リンクkがネットワークに含まれるとき$x_k=1$、含まれないとき$x_k=0$とする。

c_k：リンクkの建設費

C：ネットワーク建設費の上限値

式(6.1)は総所要時間の最小化を表わす目的関数である。そして、式(6.2)は予算制約を表わしている。

この問題を適用するときには、計画対象地域にすでに存在する交通網を考慮しなければならない。また、対象地域外の交通網との接続も考えなければならない。そこで、解のネットワークに含めるかどうかを調べる検討リンクと、つねに解のネットワークに含める既存リンクにリンクを分けて扱う。さらに、それぞれが表わすもので分類すると、リンクの種類は図−6.1のようになる。



図−6.1　リンクの分類

ダミーリンクは対象地域外との接続を表わすためのものである。既存リンクに対しては、定式化の上からは$x_k=1$の制約式を加えることになる。

問題6.1は最適交通網構成問題の基本問題と類似の形をしている。したがって、基本問題に対する解法を基礎にできる。しかし、計算を効率的に行うには、問題の特性を生かすように解法を修正しなければならない。そこで、次に示す点に関してアルゴリズムを手直しする。

この問題の特徴のひとつは、最大ネットワークにおいて、同一のノード間に複数のリンクが存在するのが一般的なことである。目的関数は総所要時間の最小化で、リンク交通量に対する容量制約はない。そのため、所要時間の短いリンクがあれば、それと両端点が同じで所要時間の長いリンクは不要である。このようなリンクをネットワークに含めても、建設費をむだに使うだけで目的関数値は改善されない。よって、所要時間の長いリンクは、それと両端点が同じで所要時間の短いリンクのすべてがネットワークに含まれないときにのみ、ネットワークに含めるかどうかを検討すればよい。ここでは鉄道、新交通システム、バスの順に速度、建設費の単価が大きいものと仮定し、不要な解の探索を省く。

鉄道や新交通システムの場合、長さの短いリンクが散在する計画は実現できない。とくに対象地域と都心を結ぶ鉄道では、都心に近いところからどれだけ対象地域の奥深く伸ばすかということになる。そこで、このような鉄道のリンクは、それよりも都心側に位置する鉄道のリンクが含まれなければネットワークに含めないことにする。新交通システムの場合はネットワークが複雑なのでこのような制約は設けず、得られた解が妥当かどうかを検討する。そして、必要ならば検討リンクを既存リンクに変更したりして再計算を行うものとする。

これらについて修正すれば、解法の基本的な部分は基本問題に対するものが使える。実用性から近似解法で解くならば、局所最適化を行うものや、リンクの段階的削減を行うものも適用できる。そこで、このふたつの近似解法を簡単な例題の計算で比較すると、局所最過化を行うものの方が計算時間が短かった。解の探索範囲を決めるパラメータの値の選択も容易なので、この解法を用いることにする。

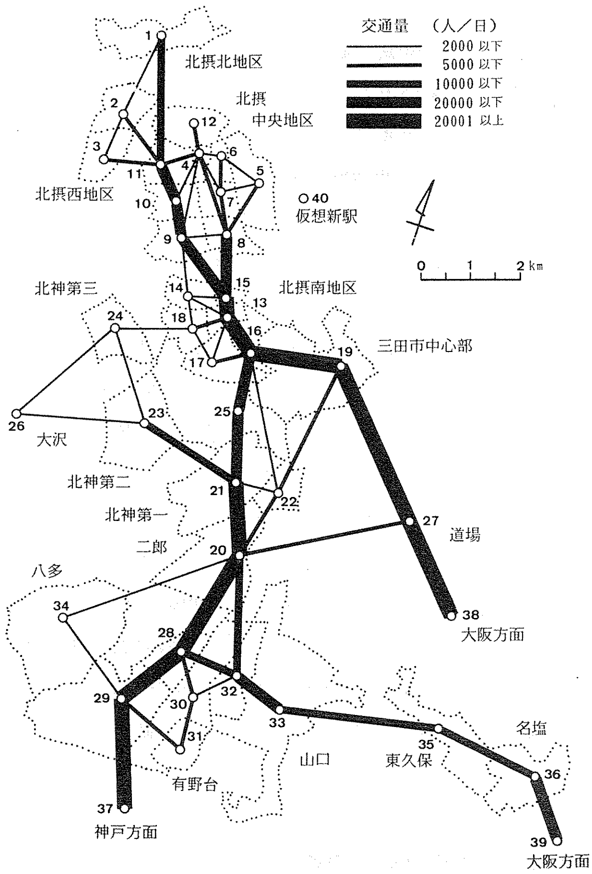
6．2．3　北摂・北神地域の交通網計画への適用

ここではケーススタディとして、北摂・北神地域の交通網計画を最適交通網構成手法を用いて行う。

計画の対象地域は兵庫県の神戸市北区、三田市、西宮市、吉川町にまたがり、北摂ニュータウンや北神ニュータウンが計画され、民間の開発も盛んな地域である。北摂・北神地域に対する交通網計画は、この地域と神戸の都心を結ぶ鉄道のルートを決め、地域内の交通を処理する大量輸送機関網を設定するものである。ここでは、このうちの地域内の交通網を決める問題を扱う。

北摂・北神地域内の交通網の可能な計画として、国鉄福知山線の電化とルートの変更、導入線と呼ぶ二郎から北摂ニュータウン内への鉄道の延長を考える。ここでの交通網計画では、これらの鉄道の建設の可否と新交通システムの導入の可能性の検討、それらを補うバスネットワークの設定を行う。

対象地域のゾーン分割や分布交通量などのデータは、兵庫県土木部が行った北摂・北神及びその周辺地域の交通体系調査作業結果を用いる。対象地域を36ゾーンに分割し、各ゾーンを代表するノードを設ける。ゾーンの境界とノードの位置、それにトラフィックラインなどを図－６．２に示す。この図におけるノード37～39は地域外との流出入をまとめるためのダミーノードである。またノード40は福知山線のルートを２種類検討するためのものである。ひとつのルートはノード19の三田駅から北摂中央地区のノード４に至るものである。もう一方は現在のルートのままで、



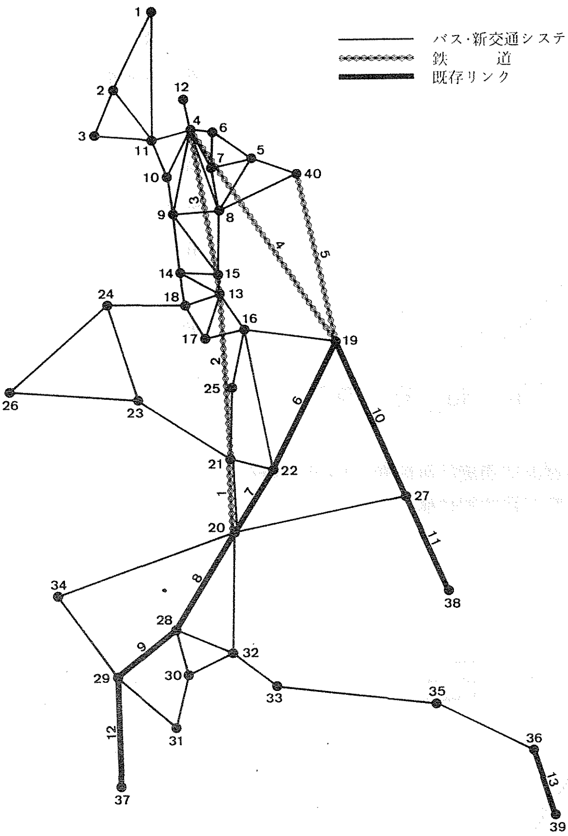
図―6.2　北摂・北神地域のゾーン，ノード，トラフィックライン

この場合には北摂ニュータウンに近い所に新駅をつくるものとし、仮想新駅に対応するノード40を設ける。

需要交通量は出勤目的の全手段に対するものを用いる。これをトラフィックラインに配分すると、図－6.2のようになる。トラフィックラインの線の太さで交通量の大きさを表わしている。これを見れば、北摂地区から大阪方面への交通量と、北神地区以南から神戸方面への交通量が多いことがわかる。

最大ネットワークはトラフィックラインを参考に図－6.3のようにする。図－6.3では鉄道リンクと既存リンクについてはリンク番号を示した。鉄道はリンク1～3で導入線を、リンク4、5で福知山線の検討対象のルートを表わす。既存リンクはリンク6～9が神戸電鉄三田線、リンク10が国鉄福知山線である。そして、リンク11～13は地域外との接続点のダミーリンクである。リンク番号を示していないバスと新交通システムのリンクは、どちらもトラフィックラインと同じパターンである。

リンクの長さはノード間の直線距離とする。これを表定速度で割ってリンクの所要時間を求める。表定速度は計画されている値、あるいは現在の値を用いる。リンクの建設費はリンクの長さ



図―6.3　北摂・北神地域の交通網計画の最大ネットワーク

に建設費の単価を掛けて求める。建設費の単価と新交通システムの表定速度は、東京50km 圏総合交通体系調査報告書[2)]と神奈川県内新交通輸送体系確立に関する報告書[3)]を参考に決める。リンクの種類ごとの表定速度と建設費単価を表－6.1に示す。なおダミーリンクの所要時間は0とする。

表―6.1　リンクの表定速度と建設費単価

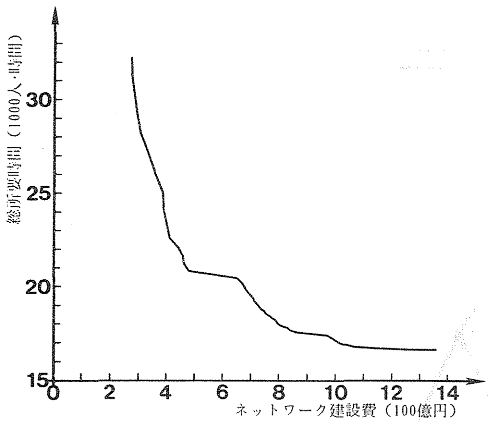
リンクの種類	表定速度 (km／h)	建設費単価 (億円／km)
国鉄福知山線	40	33
神戸電鉄三田線	31	
導入線	45	33
新交通システム	30	10
バス	12	0.12
徒歩	4.8	

導入線ではリンク1が都心側であり、これを除くときはリンク2、3もネットワークに含めない。リンク3はリンク1、2がどちらもネットワークに含まれるときにだけ、含めるかどうかを調べる。これは、式（6.4）、（6.5）の制約式を加えることに相当する。

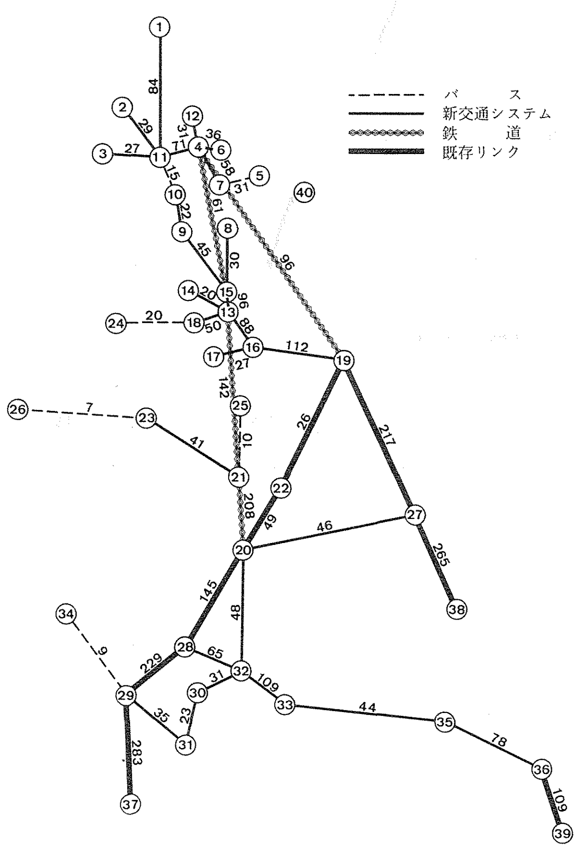
$$x_1 - x_2 \geqq 0 \qquad (6.4)$$

$$x_2 - x_3 \geqq 0 \qquad (6.5)$$

まず backward 法を用いて最大ネットワークからリンクを除いてゆき、ネットワーク建設費と総所要時間の関係を調べる。この計算の結果を図－6.4に示す。この図を見れば、ネットワー



図— 6.4　北摂・北神地域の交通網計画問題の**backward**法による近似解の目的関数値



図— 6.5　北摂・北神地域の交通網計画問題のケース１の解

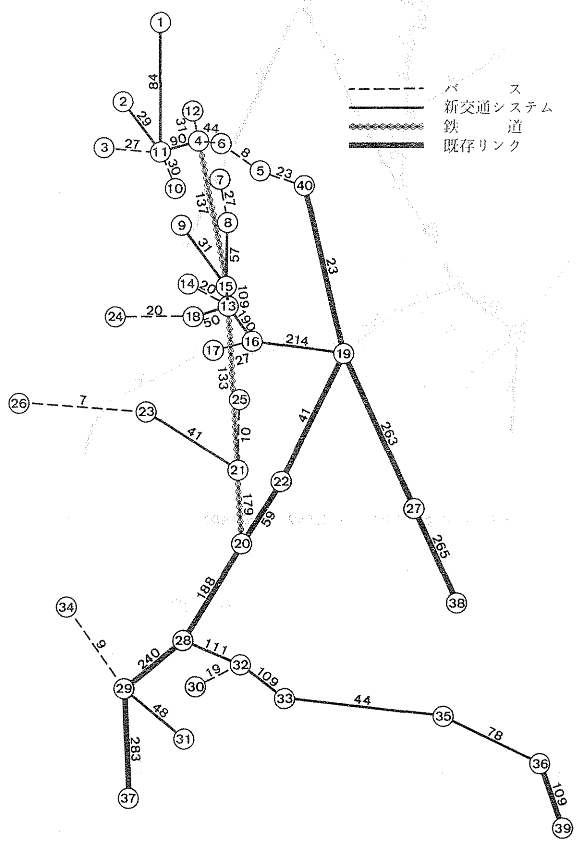
ク建設費が小さくなれば、総所要時間が急激に増加している。しかし、建設費が800億円程度であれば、最大ネットワークでの総所要時間と大きな差はない。そこで、ケース１として予算が800億円の場合の解を求める。なお、**6．3．2**で述べた近似解法を適用するとき、backward 法の解の近傍を探索する範囲を決めるパラメータの値は６とする。すなわち、最大で12本のリンクの組み合わせを調べる。

ケース１の解のネットワークは図−**6．5**に示すものである。対象地域内には大きな就業地がないので、出勤目的の需要交通量を配分すると、ほとんどのリンクは極端な片荷輸送になる。そこで、図−**6．5**には、それぞれのリンクの両方向の交通量のうち、大きい方だけを100人／日単位で示した。この解のネットワーク建設費は798億円である。福知山線は北摂中央地区へ乗り入れるルートをとり、導入線も北摂中央地区まで伸ばす計画である。そして、新交通システムを35.8km つくることになる。内々トリップを除いたトリップの平均所要時間は10.3分である。これは、内々トリップを除いたトリップが域内で完結するか、対象地域から出るまでに平均10.3分かかることを意味している。

ケース１の解でのリンク交通量を調べると、多い方向でも2025人／日というリンクが新交通システムにある。バスでは、１本のリンクを除いてそれ以下の交通量である。この地域における大量輸送機関の時間交通量の最大値は、

出勤目的の全手段の需要交通量を配分したリンク交通量の63%になると予測されている。そこで、リンク交通量2025人／日をピーク時に換算すれば1276人／時間になる。5000人／時間位まではバスで処理できるとされており、新交通システムの建設費がバスと比較にならないほど高いことを考えれば、ケース１の解は現実的でない。また福知山線はニュータウン内を通し、北摂中央地区へ乗り入れた方が良いという結果であるが、そのような建設が困難な場合が想定できる。

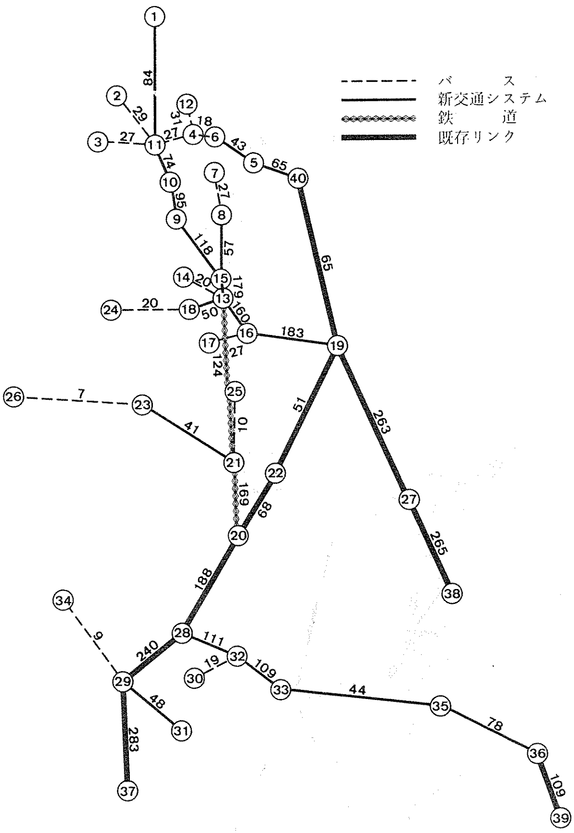
これらを考慮し、福知山線を現在のルートとし、予算額を小さくした場合の解を求める。すなわち、リンク５を既存リンクとして扱い、それ以外の予算を500億円とし、これをケース２とする。計算の結果、この解は図－６．６に示すものになった。ネットワーク建設費は福知山線を含



図― 6.6　北摂・北神地域の交通網計画問題のケース２の解

めて602億円である。導入線は北摂中央地区までで、新交通システムは22.2km である。そして、トリップの平均所要時間は11.6分である。福知山線の三田以北の利用者がケース１の11%に減少していることが目立つ。新交通システムのリンク交通量の最小値は2851人／日で、なお交通量の少ないリンクに新交通システムが使われている。

ケース２の解でも実現は困難であり、予算を減らしたために新交通システムの短い路線が散在するようになってきている。これを解決するため導入線を北摂南地区以北には伸ばさないことにし、予算を400億円とする。これをケース３として計算した結果、図－６．７の解が得られた。



図— 6.7　北摂・北神地域の交通網計画問題のケース 3 の解

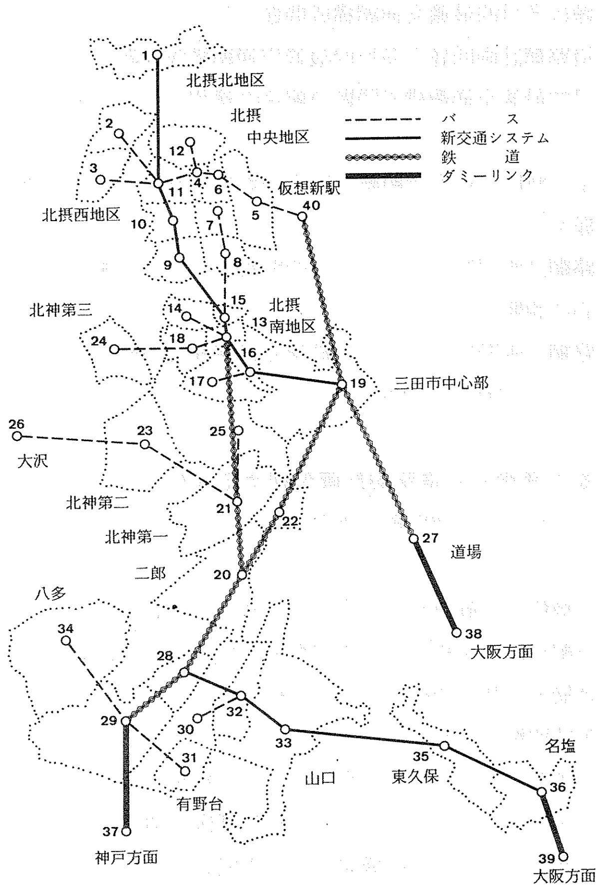
ネットワーク建設費は福知山線を含めて505億円である。導入線は北摂南地区までで、新交通システムは23.7km である。トリップの平均所要時間は11.7分であり、福知山線の三田以北の利用者数はケース 2の2.8倍である。新交通システムのリンク交通量の最小値は4128人／日である。

ケース 3 はケース 2 よりも予算を減らしたにもかかわらず、目的関数値はあまり変わらない。これは、導入線を北摂南地区までしか伸ばさない代わりに、新交通システムのリンクが多くなったからである。ケース 3 のネットワークでも新交通システムの輸送力を超える交通量はなく、この点の問題はない。そして、福知山線の三田以北の利用者が比較的多いことも好ましい。

以上の計算結果から、福知山線を現在のルートのままにするならば、可能な北摂・北神地域の交通網計画案は図－ 6 . 8 になる。すなわち、導入線を北摂南地区までとし、三田駅から北摂南地区、北摂中央地区を経て北摂北地区までと、名塩から東久保、山口を経て田尾寺までの新交通システムを建設するものである。

6 . 2 . 4　むすび

この節では、鉄道、新交通システム、バスの 3 種類の大量輸送機関を組み合わせ、地域の交通網を計画する問題を取り上げた。最初に、この問題を解くのに必要な、最適交通網構成問題の基



図― 6.8　北摂・北神地域の現実的な交通網計画案

本問題の近似解法の修正について述べた。そして、北摂・北神地域の交通網計画の問題をケーススタディとして解いた。

最適交通網構成問題は問題を単純化しているため、定式化に組み込めない評価項目や制約条件がある。ケーススタディでは得られた解を検討し、必要な修正を加えて再計算することでこれを解決した。ここで対象とした問題は基本問題に近く、最適交通網構成問題のなかでは簡単なものである。しかし、ケーススタディによってその実用性は実証された。

6.3　道路網計画問題への適用[4)~6)]

6.3.1　はじめに

道路網計画は交通網計画の代表的なものである。従来は設定した需要交通量に対応できる機能を持つ道路網を構成することだけが考えられていた。しかし、自動車交通が沿道の環境に及ぼす影響が問題にされるようになり、騒音や排出ガスなどの環境要因をも考慮して道路網を計画することが必要になった。そこで、この節では環境要因をも考慮して道路網を計画する問題に最適交通網構成手法を適用する。道路網の機能を表わす評価指標に加え、環境要因を表わす評価指標を

用いるため、この問題は多目的最適交通網構成問題になる。

最初に対象とする道路網計画問題を多目的最適交通網構成問題として定式化し、目的関数を説明する。つぎに、多目的最適交通網構成問題の解法の適用について述べ、ケーススタディを行う。

6.3.2では、総走行時間、総走行距離、騒音の影響、排出ガスの影響の最小化を目的関数とする道路網計画問題を示す。

6.3.3では、道路網計画問題で採用した目的関数について、騒音と排出ガスの環境要因の予測と評価方法を中心に説明する。

6.3.4では、道路網計画問題への多目的最適交通網構成問題の解法の適用について述べる。

6.3.5では、神戸市の道路網を対象にケーススタディを行う。

6.3.2 環境要因をも考慮した道路網計画問題の定式化

ここでは、この節で対象とする道路網計画問題について述べ、これを多目的最適交通網構成問題として定式化する。

道路網に要求される機能は自動車交通を円滑に流せることである。この機能を確保するため、設定した需要交通量の配分結果を検討し、道路網案を採択する計画手法がとられる。この場合の道路網の評価指標には総走行時間や総走行距離が使われている。そして、実際の交通流に近い状態で評価するために等時間配分が多く用いられる。

けれども、交通量の増加によって沿道の環境へ及ぼす影響を無視できなくなり、機能だけを考えて道路網を計画することは許されなくなった。環境要因を道路網の評価指標に加えることが必要になったのである。そこで、ここでは総走行時間と総走行距離のほかに、代表的な環境要因である騒音と排出ガスの影響を考慮して道路網を計画する問題を取り上げる。

この道路網計画問題に最適交通網構成手法を適用するならば、4種類の目的関数を同時に考えなくてはならない。すなわち、多目的最適交通網構成問題となるのである。制約条件はネットワーク建設費の予算制約だけとすると、問題は次のように表わせる。

問題6.2

$$\min \quad f_i(\boldsymbol{x}) \quad (i=1,2,3,4) \tag{6.6}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} c_k x_k \leq C \tag{6.7}$$

$$x_k = 0 \ \mathit{or} \ 1 \quad (k=1,2,\cdots,m) \tag{6.8}$$

ここに、f_1：総走行時間

f_2：総走行距離

f_3：騒音による影響

f_4：排出ガスによる影響

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_m\}$

：ネットワークの状態を表わし、リンクkがネットワークに含まれるとき$x_k=1$、含まれないとき$x_k=0$とする。

c_k：リンク k の建設費

C：ネットワーク建設費の上限、すなわち予算

式（6．6）は目的関数ベクトルの最小化を表わす。式（6．7）はネットワーク建設費の予算制約である。目的関数のそれぞれについては後で詳しく述べることにし、ここでは問題全体の構成を示すだけにする。

目的関数の値はネットワークにより変化するが、その値を求めるには交通配分方法が決まっていなくてはならない。環境要因を考慮して道路網計画を行うとき、環境への影響を最小にする交通配分方法を使っている研究がある。[7)8)] しかし、自動車運転者が沿道の環境を考慮して経路を選択するという仮定は現実的でない。そのような交通配分を交通規制で実現するとしても、有効な規制方法があるか疑わしい。そこで、交通配分方法は実際の交通流に近い配分結果が得られるとされる等時間配分を用いる。

等時間配分を用いることは、利用者が通常の経路選択を行った場合において、最適な状態になるように道路網を決めることを意味している。そして、等時間配分の前提として、各リンクの所要時間は交通量によって変化するものとする。これは、環境要因を目的関数に加えるためにも必要である。

ここで定義した道路網計画問題の最適交通網構成問題としての特徴は、次の2点である。

① 多目的最適交通網構成問題である。

② 等時間配分を用いる。

6．3．3　道路網計画問題に用いる目的関数

ここでは、道路網計画問題の目的関数に選んだ総走行時間、総走行距離、騒音の影響、排出ガスの影響について述べる。

総走行時間と総走行距離は、道路網の評価にもっとも多く使われているものである。最適交通網構成問題でもよく使われている。ここで対象とする問題6．2では等時間配分を行うことを前提としている。そのため、同一のノード間の所要時間は一意に定まるが、走行距離はトリップの経路により異なる。そこで、これまでの最適交通網構成問題のようにノード対ごとの走行距離を合計するのでなく、リンクごとの値を合計するように表わした方がよい。総走行時間もこれにあわせ、それぞれ式（6．9）、（6．10）とする。

$$f_1(\boldsymbol{x})=\sum_{k=1}^{m}\sum_{h=1}^{2} t_{kh}\, q_{kh} \qquad (6.9)$$

$$f_2(\boldsymbol{x})=\sum_{k=1}^{m}\sum_{h=1}^{2} l_k q_{kh} \qquad (6.10)$$

ここに、t_{kh}：リンク k の方向 h の所要時間

q_{kh}：リンク k の方向 h の交通量

l_k：リンク k の長さ

リンクの交通量はネットワークに需要交通量を等時間配分して求める。そして、リンクの所要時間は、式（6．11）により交通量の関数として決まるものとする。

$$t_{kh}=g_k(q_{kh}) \qquad (6.11)$$

ここに、g_k：リンク k の交通量と所要時間との関係を表わす関数

騒音や排出ガスが沿道に及ぼす影響を表わす目的関数をつくるにあたり、これらの予測式を定めなければならない。騒音については、日本音響学会の予測計算方法に準じて行う[9]。すなわち、等間隔モデル式を用い、平均パワーレベルは２車種区分の式で計算する。ただし、簡単のために平均車頭間隔に対する受音点までの距離の比が⅙よりも大きい場合の式だけを使う。そして、２方向の交通による騒音を簡便に合成するため、平均車頭間隔の逆数は各方向の平均車頭間隔の逆数の和とし、パワーレベルは両方向の平均値を用いる。さらに、市街地の道路を対象とするので、沿道の建物による音の減衰の補正を行う。これには、加来・山下が提案している式[10]を用いる。そうすると、リンクの沿道の騒音レベル中央値は式（６.12）で表わされる。

$$E_1^k(y) = 54 + 0.1(v_{k1} + v_{k2}) + 10\,log(a_1^k + 10\,a_2^k) + 10\,log\left(\frac{q_{k1}}{v_{k1}} + \frac{q_{k2}}{v_{k2}}\right) - 10\,log\ y - \frac{7.6\,N_k^{0.8}}{H_k + 20}\left\{1 - exp\left(-\frac{y}{75}\right)\right\} \tag{6.12}$$

ここに、 E_1^k：リンク k の沿道の騒音レベル中央値（dB（A））

y ：道路の中心から受音点までの距離（m）

v_{kh}：リンク k の方向 h の走行速度（km／h）

a_1^k ：リンク k の小型車混入率

a_2^k ：リンク k の大型車混入率

q_{kh}：リンク k の方向 h の交通量（台／h）

H_k：リンク k の高架高さ（m）

N_k：リンク k の沿道の家屋密度（戸／（200m×200m））

排出ガスは、走行速度と排出ガス濃度の関係が比較的はっきりしており、自動車以外が原因のものについては環境基準値以下である一酸化炭素だけを対象とする。排出ガス濃度の予測モデルは騒音の場合ほど確立されていない。また実測値とよく合うモデルであっても、複雑な計算を要する難点がある。そこで、簡単なモデルである船舶技術研究所の Pasquill 式変形モデル[9]を用いる。そして、この式で使われる汚染物質源の強さは、阪神高速道路公団の研究による走行速度と汚染物質の排出量の関係式[11]を用いて算定する。そうすると、汚染源を自動車交通だけとし、撹拌混合高さ以内について考える場合には、沿道の排出ガス濃度の予測式は式（６.13）のようになる。ただし、道路端撹拌混合層高さは一般に用いられる２mとし、汚染物質源の強さは２方向の交通について合成する。

$$E_2^k(y) = \frac{511.12\,(v_{k1}^{-1.125}\,q_{k1} + v_{k2}^{-1.125}\,q_{k2})}{4500\{2 + (0.234\,y)^{0.7}\}} \tag{6.13}$$

ここに、E_2^k：リンク k の沿道の排出ガス濃度（ppm）

y ：道路端からの距離（m）

u ：風速（m／s）

騒音や排出ガスが沿道に及ぼす影響の大きさの表現方法は、必ずしも確立されていない。ここでは、これらの環境要因の予測値と環境基準値との差と地域の人口の積和を環境評価値とする。

そして、リンクの環境評価値の総和を目的関数とする。これを式で表わすと次のようになる。

$$f_3^k = \begin{cases} p_k l_k \int_{y_0^k}^{y_1^k} \{E_1^k(y) - E_1^{0k}\} dy & \\ & if \ E_1^k(y_0) > E_1^{0k} \\ 0 & if \ E_1^k(y_0) \leqq E_1^{0k} \end{cases} \tag{6.14}$$

$$f_4^k = \begin{cases} p_k l_k \int_0^{y_2^k} \{E_2^k(y) - E_2^{0k}\} dy & \\ & if \ E_2^k(0) > E_2^{0k} \\ 0 & if \ E_2^k(0) \leqq E_2^{0k} \end{cases} \tag{6.15}$$

$$f_3(\boldsymbol{x}) = \sum_{k=1}^{m} f_3^k \tag{6.16}$$

$$f_4(\boldsymbol{x}) = \sum_{k=1}^{m} f_4^k \tag{6.17}$$

ここに、 f_3^k ：リンク k の騒音評価値

f_4^k ：リンク k の排出ガス評価値

p_k ：リンク k の沿道の人口密度

E_1^{0k}：リンク k の沿道の騒音基準値

E_2^{0k}：リンク k の沿道の排出ガス基準値

y_0^k ：リンク k の道路の中心から道路端までの距離

y_1^k ：リンク k の騒音が基準値を満たす地点までの道路の中心からの距離

y_2^k ：リンク k の排出ガスが基準値を満たす地点までの道路端からの距離

式（6.14）はリンクの騒音評価値を定める式である。騒音の予測式では道路の中心を距離の原点としたので、積分の下限値を道路端までの距離にしている。式（6.15）はリンクの排出ガス評価値を定める式である。そして、式（6.16）、（6.17）はそれぞれ騒音と排出ガスの影響を表わす目的関数である。

6.3.4 道路網計画問題の解法

多目的最適交通網構成問題の計算手順は**5.3**で提案した。ここでは、問題6.2の道路網計画問題をこの手順で解くとき、問題に合わせて直さなければならない部分について述べる。そして、使用する多目的決定問題の最適化手法を示す。

5.3.4で提案した多目的最適交通網構成問題の計算手順では、まず近似解を求めて検討リンクを決定する。つぎに近似解の近傍で実行可能解を列挙、限定し、そのうちの非劣解を選び出して多目的決定問題の最適化手法を適用する。問題6.2は、この計算手順の作成の対象とした問題5.3と同じ形なので、これをそのまま用いることができる。

この計算手順で問題6.2を解くときに検討を要するのは交通配分である。この問題では等時間配分を行って道路網を評価することを前提としている。しかし、等時間配分には計算時間がかかる。そのため、目的関数値を計算するたびに等時間配分を行うのであれば、小規模の問題しか

解けない。そこで、多数の解の目的関数を繰り返し求めるときには、最短経路配分を用いなくてはならない。その場合でも、最終的には等時間配分を行って道路網を決めることを考慮しておかなければならない。

近似解は総走行時間の最小化を目的関数とし、backward 法で求める。リンクの所要時間を固定し、最短所要時間の経路へ交通配分して目的関数値を計算する。そして、リンクを１本除くごとに、配分した交通量をもとにリンクの所要時間を求め直す。また、交通量が交通容量を超えるリンクがあるとき、そのようなリンクは等時間配分を用いる段階で採否を決めるものとし、近似解に含める。

近似解の近傍で検討リンクを決めるときには等時間配分を用いる。実行可能解を限定するときは、多数の解の目的関数値を計算しなくてはならない。そこで、近似解を求めるときと同じように総走行時間を目的関数とする。このときの実行可能解は、近似解に近いものである。よって、近似解に含まれるリンクの所要時間には、近似解で等時間配分したときの値を用いる。それ以外の検討リンクの所要時間は、検討リンクにするかどうかを決めるときに行った等時間配分での値とする。

列挙、限定した実行可能解のなかから非劣解を選び出すときは、等時間配分を行って総走行時間、総走行距離、騒音、排出ガスの目的関数値を計算する。

非劣解集合を求めたあと、多目的決定問題の最適化手法を適用する。これには目的関数のスカラ化手法が使用できる。ここでは代表的なものとして、加重和最小化手法、ε－制約式手法、最大成分最小化手法を用意する。

加重和最小化手法[12]は、各目的関数の加重和を最小にする解を求めるものである。ここでは、次の問題を解くことになる。

補助問題６．２．１

$$\min \quad Z=\sum_{i=1}^{4} w_i f_i(\boldsymbol{x}) \tag{6.18}$$

$$\text{s.t.} \quad \boldsymbol{x} \in X \tag{6.19}$$

ここに、　w_i：目的関数 i に対する重み

　　　　　X：求めた非劣解集合

ε－制約式手法[12]は、あるひとつの目的関数だけを残し、それ以外の目的関数を制約式に置き換えるものである。ここでは、次の問題を解くことになる。

補助問題６．２．２

$$\min \quad Z=f_p(\boldsymbol{x}) \tag{6.20}$$

$$\text{s.t.} \quad f_i(\boldsymbol{x}) \leqq \varepsilon_i \qquad (i=1,\ 2,\ 3,\ 4,\ i \neq p) \tag{6.21}$$

$$\boldsymbol{x} \in X \tag{6.22}$$

ここに、　f_p：選ばれた目的関数

　　　　　ε_i：目的関数 i の許容限界

最大成分最小化手法[12]は、目的関数のなかで最大のものが最小である解を求めるものである。

ここでは、次の問題を解くことになる。

補助問題６．２．３

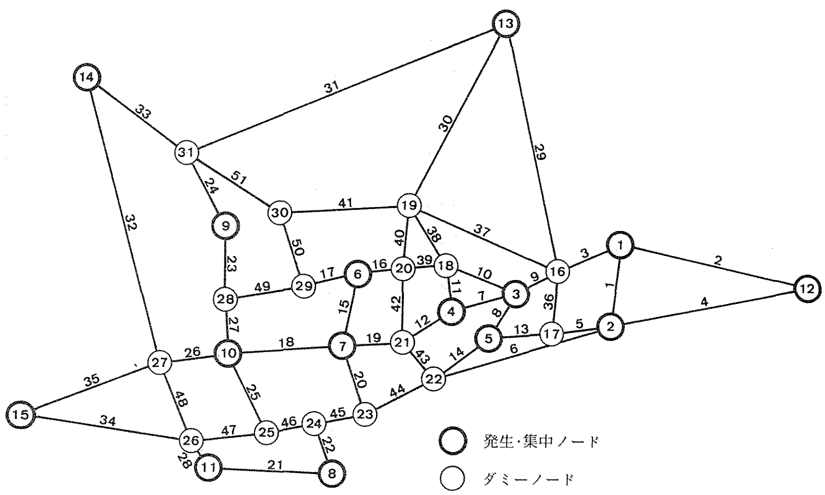
$$\min \quad Z = \max_{i} \{ w_i f_i(\boldsymbol{x}) \} \tag{6.23}$$

$$\text{s.t.} \quad \boldsymbol{x} \in X \tag{6.24}$$

６．３．５　神戸市の道路網への適用

ここでは、神戸市の道路網を対象とし、多目的最適交通網構成問題の解法を用い、環境要因をも考慮した道路網計画のケーススタディを行う。

対象とする地域は神戸市の中央区、兵庫区、長田区である。地域内に11ゾーンを設け、地域外は４ゾーンにまとめる。そして、平面街路の主要な路線を選び、建設が可能な道路区間を加えて51リンクを検討対象とする。最大ネットワークは図－６．９に示すもので、リンクの接続を表わすためにダミーノードを設けている。



図― 6.9　神戸市の道路網計画問題の最大ネットワーク

リンクは高規格道路とその他の一般道路を表わすものに分ける。そして、道路状況に応じて交通容量を設定する。建設費は表－６．２の単価をもとに想定する。騒音の基準値は道路に面する地域の朝夕の時間区分の環境基準値を用いる。排出ガスは一酸化炭素を対象としているが、一般には道路環境基準の値を超える区間はほとんどない。そこで、排出ガスによる影響を調べるため、とくに５ppmを便宜上の基準値とする。家屋密度は沿道の区域の値を面しているリンクの長さで加重平均して求める。リンクに関するデータをまとめたのが表－６．３である。

表― 6.2　リンクの建設費単価

交通容量	建設費単価(億円/km)	
(台／h)	一般道路	高規格道路
2400以下	20	50
4800以下	28	70
7200以下	36	90
7201以上	44	110

表—6.3　神戸市の道路網計画問題のリンクに関するデータ

リンク	端	点	リンク長	建設費	道路規格	交通容量	騒音基準値	排出ガス基準値	人口密度	家屋密度
1	1	2	0.9	18	2	1800	55	5	50	461
2	1	12	2.3	207	1	6000	55	5	50	600
3	1	16	1.3	91	1	4800	65	5	40	311
4	2	12	2.1	231	1	10000	65	5	10	400
5	2	17	1.5	135	1	7200	65	5	5	93
6	2	22	3.0	210	1	4000	65	5	1	40
7	3	4	1.2	84	1	4000	65	5	15	365
8	3	5	1.0	20	2	1800	65	5	15	364
9	3	16	0.7	49	1	4000	65	5	15	461
10	3	18	1.4	28	2	800	50	5	15	276
11	4	18	0.6	12	2	1200	50	5	15	326
12	4	21	1.0	70	1	4000	55	5	20	318
13	5	17	1.0	70	1	6000	65	5	5	62
14	5	22	1.2	108	1	5000	65	5	5	84
15	6	7	1.3	26	2	1200	55	5	45	538
16	6	20	1.0	20	2	800	50	5	35	478
17	6	29	1.0	20	2	800	50	5	55	462
18	7	10	2.1	189	1	6400	55	5	50	494
19	7	21	0.6	54	1	6400	65	5	25	296
20	7	23	1.0	20	2	2400	65	5	20	378
21	8	11	2.9	58	2	2400	65	5	30	310
22	8	24	1.2	43	2	6000	65	5	15	310
23	9	28	1.2	24	2	2400	55	5	10	199
24	9	31	2.6	52	2	2400	55	5	2	80
25	10	25	0.9	25	2	3200	65	5	15	383
26	10	27	1.2	108	1	6000	65	5	30	426
27	10	28	1.2	24	2	2400	55	5	25	501
28	11	26	0.6	17	2	4800	65	5	50	549
29	13	16	8.0	160	2	1200	50	5	1	40
30	13	19	7.2	144	2	800	50	5	2	80
31	13	31	9.1	455	1	2400	55	5	10	400
32	14	27	5.1	102	2	1200	55	5	5	200
33	14	31	1.6	32	2	1800	50	5	2	120
34	15	26	3.3	297	1	6000	60	5	35	400
35	15	27	3.1	217	1	4000	60	5	30	400
36	16	17	1.0	20	2	1800	65	5	10	360
37	16	19	2.8	56	2	800	60	5	1	40
38	18	19	1.2	24	2	800	50	5	1	40
39	18	20	0.8	16	2	800	50	5	35	416
40	19	20	1.1	22	2	800	50	5	30	90
41	19	30	2.0	40	2	800	60	5	1	40
42	20	21	1.2	24	2	1200	55	5	40	549
43	21	22	1.0	70	1	4000	65	5	45	236
44	22	23	0.9	81	1	5000	65	5	5	624
45	23	24	1.5	135	1	6400	60	5	15	300
46	24	25	0.6	42	1	4800	65	5	15	157
47	25	26	1.2	84	1	4800	65	5	25	521
48	26	27	0.8	16	2	1800	65	5	30	425
49	28	29	1.1	22	2	600	50	5	40	327
50	29	30	1.4	28	2	600	50	5	10	125
51	30	31	2.3	46	2	1200	55	5	3	160

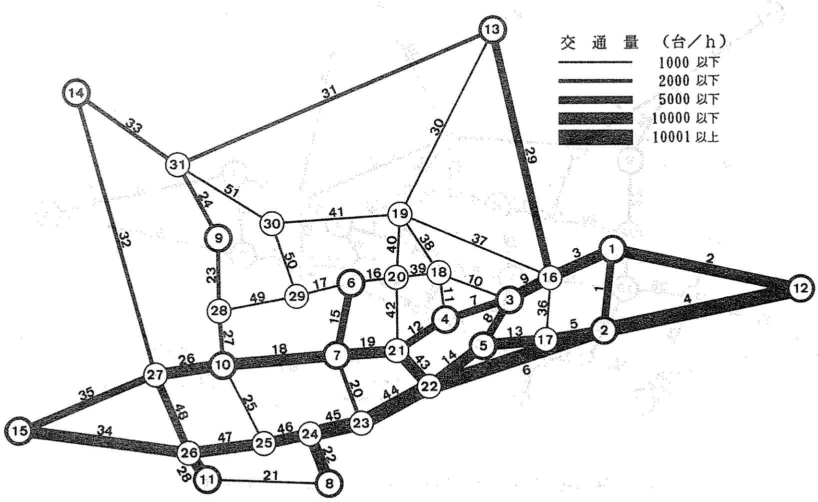
リンク長：km　　交通容量：台／h　　家屋密度：戸／(200 m×200 m)
建設費：億円　　騒音基準値：dB(A)
道路規格：1…高規格　　排出ガス基準値：ppm
2…一　般　　人口密度：1000人／km²

需要交通量は、京阪神都市圏パーソントリップ調査に基づいて神戸市都市計画局が推計した昭和65年の自動車類OD表を用いて作成する。高速道路の通過分を除き、ピーク率を6％としてピーク時の時間交通量を求める。これを示したのが表－6．4である。騒音評価値を求めるには小型

表― 6. 4　神戸市の道路網計画問題の需要交通量　（単位:台/h）

発生	集									中					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	合 計
1	0	632	88	74	347	28	45	94	10	77	51	801	92	70	308	2717
2	683	0	248	207	1036	67	129	299	24	208	163	1818	241	178	502	5803
3	90	238	0	140	658	37	69	157	8	78	67	550	85	56	288	2521
4	75	199	140	0	549	31	58	131	7	65	56	459	70	47	239	2126
5	366	1031	681	568	0	158	325	773	33	335	303	1845	373	236	782	7809
6	22	56	33	27	140	0	139	242	8	72	55	171	54	54	157	1230
7	44	128	72	60	326	171	0	511	15	157	146	433	116	97	340	2616
8	103	207	175	146	816	340	571	0	33	382	382	827	270	189	519	4960
9	5	15	34	4	25	5	11	25	0	88	77	31	9	18	61	408
1 0	53	168	66	55	302	57	136	354	93	0	1266	450	113	169	836	4118
1 1	26	169	68	56	317	54	150	400	85	1329	0	346	114	146	681	3941
1 2	782	1817	533	442	1823	201	442	816	41	471	324	0	579	254	1800	10325
1 3	59	183	69	58	345	43	95	234	6	92	102	521	0	70	330	2207
1 4	52	151	48	40	234	53	84	171	16	157	138	269	74	0	0	1487
1 5	229	455	236	196	720	127	310	509	43	768	695	1714	343	0	0	6345
合計	2589	5449	2491	2073	7638	1372	2564	4716	422	4279	3825	10235	2533	1584	6843	58613

車と大型車を分けなければならない。しかし、簡単のために配分結果において、一般道路で10％、高規格道路で20％が大型車であると仮定する。需要交通量を最大ネットワークに最短経路配分すると、図－6．10のようになる。



図― 6.10　神戸市の道路網計画問題の最大ネットワークへの配分結果

リンクの交通量と所要時間の関係を表わす式には Steenbrink の提案する式[13]を用いる。そして、一般道路と高規格道路に対して係数を決め、それぞれ式(6.24)、(6.25)を使う。

$$t = 0.031 + 0.036\left(\frac{q}{c}\right)^5 \qquad (6.24)$$

$$t = 0.021 + 0.046\left(\frac{q}{c}\right)^5 \qquad (6.25)$$

ここに、 t：走行所要時間（h／km）

q：交通量（台／h）

c：交通容量（台／h）

近似解の近傍で検討リンクを決めるには、式(5.23)を使う。近似解でリンクの状態を変えたときの目的関数値の変化を調べるのである。このとき、目的関数間のスケールを調整するための係数を決めなくてはならない。ここでは、それぞれの目的関数における最大の変化量でスケールを調整する。すなわち、式(6.26)で係数を定める。これを用いれば、リンクの状態を変えたときの各目的関数の値の変化を0と1の間の数に変換することになる。

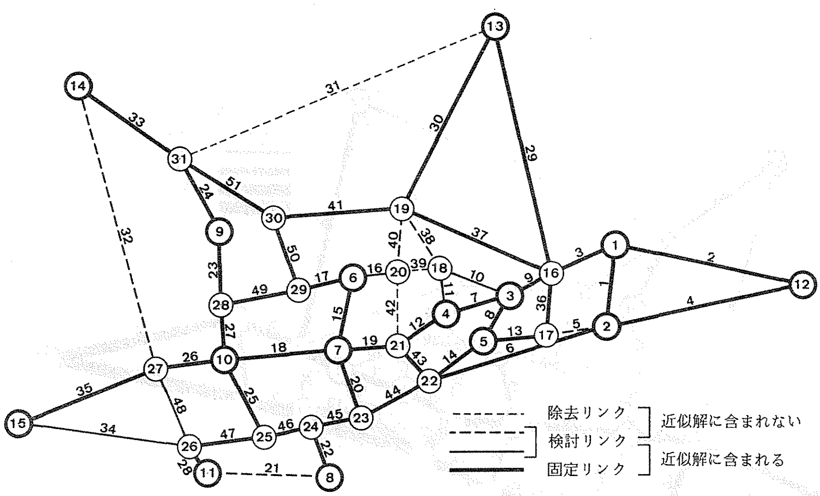
$$\alpha_i = \min_k \left\{ \frac{1}{|f_i(\boldsymbol{x}_k) - f_i(\boldsymbol{x})|} \right\} \qquad (6.26)$$

ここに、α_i：目的関数間のスケールを調整するための、目的関数 i に対する係数

$\boldsymbol{x}$：近似解のネットワーク

$\boldsymbol{x}_k$：$\boldsymbol{x}$ のネットワークにおいて、リンク k の状態を逆に変えたネットワークを表わす。

予算は最大ネットワークの建設費の80%とし、最初に backward 法で近似解を求める。そして、検討リンクは近似解に含まれるリンクと含まれないリンクから5本ずつを選ぶことにする。そうすると、図－6.11のようにリンクを分類できる。



図― 6.11　神戸市の道路網計画問題のbackward法による近似解とリンクの分類

検討リンクが定まったので、つぎにそれらの組み合わせで実行可能解を列挙する。そして、各実行可能解の目的関数値を計算し、非劣解を選び出す。その結果、170個の実行可能解を列挙し、**表－６．５**に示す19個の非劣解が得られた。それらの目的関数値は**表－６．６**のとおりである。

表— 6.5　神戸市の道路網計画問題の非劣解

解	検討リンク									
	21	32	40	5	38	48	11	10	16	34
1			○			○			○	○
2					○	○	○			○
3					○	○	○	○		○
4	○					○				○
5	○		○	○	○	○	○	○	○	
6	○		○	○	○	○	○		○	
7			○	○	○	○	○	○	○	
8	○		○	○	○	○		○	○	
9			○	○	○	○	○		○	
10	○		○	○		○			○	
11	○			○	○	○	○	○		
12	○	○		○	○	○	○	○		
13	○			○	○	○	○			
14			○	○		○	○	○	○	
15	○	○		○	○	○	○			
16				○	○	○	○			
17	○			○	○	○		○		
18		○		○	○	○	○			
19	○			○		○	○	○		

表－６．６には目的関数値とともに、170個の実行可能解での最小値が０、最大値が１になるように変換した値も示した。また、目的関数値の差が小さいため、表の数字では同じ値になっているものもある。

求めた非劣解集合に対し、多目的決定問題の最適化手法を適用する。このとき、目的関数値そのものではなく、０から１の値に変換したものを用いることにする。

加重和最小化手法では、すべての目的関数に対する重みを等しくすると、解２が選好解として得られる。そして、目的関数に対する重みを変えても極端なものでなければ、解２が選ばれることが**表－６．６**からわかる。

ε－制約式手法では、騒音と排出ガスの評価値の最小化を目的関数にした場合の解を求める。他の目的関数の許容限界はすべて等しくし、0.1、0.2、0.3、．．．、0.9と与えると、それぞれの場合の選好解は**表－６．７**のようになった。いずれの場合も許容限界が0.4以下のときには実行可能解がない。他の目的関数を考慮しないのに近い状態でなければ、解２、３、４のどれかが選ばれることがわかる。

最大成分最小化手法で各目的関数に対する重みを等しくすると、解２が選好解として得られた。すなわち、解２は４種類の目的関数に極端に大きな値を持たない解である。

表－6.6　神戸市の道路網計画問題の非劣解の目的関数値

解	目的関数値			
	総走行時間	総走行距離	騒　音	排出ガス
1	2.7897 (0　　　)	3.6744 (0.5043)	2.6890 (0.7296)	0.1927 (0.0500)
2	2.7899 (0.0001)	3.6605 (0.4218)	2.4375 (0.1734)	0.1707 (0.0340)
3	2.7970 (0.0028)	3.6641 (0.4430)	2.4510 (0.2031)	0.1550 (0.0225)
4	2.8649 (0.0290)	3.6953 (0.6284)	2.4442 (0.1882)	0.1241 (0　　　)
5	3.9343 (0.4406)	3.5917 (0.0130)	2.5930 (0.5172)	1.2996 (0.8570)
6	3.9378 (0.4420)	3.5895 (0　　　)	2.5725 (0.4719)	1.2997 (0.8571)
7	3.9463 (0.4452)	3.5949 (0.0320)	2.5945 (0.5206)	1.2995 (0.8570)
8	3.9472 (0.4456)	3.6062 (0.0991)	2.6532 (0.6504)	1.2994 (0.8569)
9	3.9533 (0.4480)	3.5967 (0.0428)	2.5672 (0.4601)	1.2996 (0.8570)
10	3.9685 (0.4538)	3.6084 (0.1122)	2.6324 (0.6045)	1.2995 (0.8570)
11	3.9691 (0.4540)	3.5979 (0.0501)	2.3795 (0.0451)	1.2998 (0.8572)
12	3.9828 (0.4593)	3.5957 (0.0368)	2.3824 (0.0515)	1.3002 (0.8575)
13	3.9838 (0.4597)	3.5992 (0.0577)	2.3631 (0.0087)	1.2998 (0.8572)
14	3.9859 (0.4605)	3.6139 (0.1450)	2.6296 (0.5982)	1.2995 (0.8570)
15	3.9904 (0.4622)	3.5987 (0.0546)	2.3674 (0.0183)	1.3002 (0.8575)
16	3.9983 (0.4652)	3.6060 (0.0982)	2.3592 (0　　　)	1.2998 (0.8572)
17	4.0060 (0.4682)	3.6237 (0.2031)	2.4136 (0.1204)	1.2996 (0.8571)
18	4.0070 (0.4686)	3.6040 (0.0863)	2.3611 (0.0043)	1.3001 (0.8574)
19	4.0324 (0.4784)	3.6327 (0.2566)	2.3870 (0.0616)	1.2996 (0.8570)

総走行時間：10^4台・時間　　騒　音：10^5人・dB(A)
総走行距離：10^5台・km　　排出ガス：10^6人・ppm

表－6.7　ε－制約式手法の適用結果

許容限界	選好解	
	騒音最小化	排出ガス最小化
0.1	—	—
0.2	—	—
0.3	—	—
0.4	—	—
0.5	2	3
0.6	2	3
0.7	2	4
0.8	2	4
0.9	16	4

これらの計算結果から、総走行時間、総走行距離、騒音評価値、排出ガス評価値を目的関数とするとき、多くの場合に解２が選ばれると考えられる。そして、**表－６.６**からは、比較の対象とする解は、ほぼ解１～４に限られることがわかる。**表－６.５**の検討リンクは、検討リンクを選ぶときの評価値の順に並べてある。この表で解１～４を見ると、これらの解は backward 法の近似解に近いものであることがわかる。これは、近似解を求め、検討リンクを定める方法が妥当であったことを示している。

６.３.６　むすび

ここでは、環境要因をも考慮して道路網を計画する問題を扱った。道路網の機能を表わす総走行時間と総走行距離に加え、騒音と排出ガスの評価値を目的関数として多目的最適交通網構成問題を定式化した。そして、近似解の近傍で検討リンクを定め、その組み合わせでできる実行可能解から非劣解を選び出す解法の適用について述べた。

ケーススタディとして神戸市の道路網へ適用した結果、ここで示した解法の実用性が確められ

た。対象とした問題の特徴のひとつは多目的最適交通網構成問題であることである。これに関しては、あらかじめ非劣解集合を求めておくことにより、選好解の決定を容易に行うことができた。また、最適交通網構成問題で等時間配分を用いるのは、計算時間が長くなるために困難であった。ここでは、限られた計算過程でだけ等時間配分することにし、これを解決した。

ここで示した道路網計画問題の解法は、複数の目的関数を考慮し、等時間配分を行うという複雑な最適交通網構成問題を、実用的な水準で解けることに意義がある。実用性を重視するために簡略化を行い、解の厳密さは失われている。しかし、最適交通網構成問題を解くのに必要な計算量を考えれば、このような解法をとることはやむを得ないであろう。

6．4　道路網の対災害信頼性の最適化問題への適用[14)]

6．4．1　はじめに

地震や風水害、あるいは積雪などの大規模な自然災害がある地域を襲ったとき、その地域の道路網がなお健全に機能し続けることが望ましい。しかし、災害に対するネットワークとしての信頼性を考慮した道路網計画はみられない。社会における道路交通の重要性からすれば、災害時における状態をも考慮して道路網を計画することが必要である。そこで、ここでは道路網の災害に対する信頼性を最大にするように、各リンクの対災害信頼性の水準を最適に配分する問題を取り扱う。

道路網の災害に対する信頼性に関しては、ほとんど研究されていない。したがって、道路網の対災害信頼性の評価方法から検討を始めなければならない。そして、その評価方法に基づいて最適交通網構成問題として定式化し、解法を提案する。

6．4．2では、道路網の対災害信頼性の概念について考察する。まず、災害時の道路網をどう評価するかを検討する。そして、道路網計画問題において、災害やリンクの破壊をどのように扱うかを考える。

6．4．3では、ある災害に対して道路網の破壊パターンがひとつに決まると仮定し、そのときの総所要時間の最小化を目的関数とする問題を示す。

6．4．4では、災害時の総所要時間を最小化する問題の解法を提案する。

6．4．5では、災害時の総所要時間を最小化する問題の計算例を示す。

6．4．6では、ある災害に対して道路網の状態が確率的に決まると仮定し、目的地へ達することができるトリップ数の期待値の最大化を目的関数とする問題を示す。

6．4．7では、災害時の到達トリップ数を最大化する問題の解法を提案する。

6．4．8では、災害時の到達トリップ数を最大化する問題の計算例を示す。

6．4．2　道路網の対災害信頼性の概念

道路網に災害が加わり、一部のリンクが使用できなくなれば、道路網の状態は平常時とは異なる。使用できないリンクが生じれば、トリップの所要時間が増加し、利用者は不利益を被むる。甚だしいときには道路網が非連結になり、目的地へ到達できないトリップがあることも考えられる。そこで、対災害信頼性の高い道路網は、平常時だけでなく災害時にもノード間に経路が存在し、その所要時間が短いものとする。

ここでは、リンクに損傷を与えて使用不能にする可能性のあるものの総称を災害としている。災害によってリンクが破壊するか否かは、リンクの強度と災害の強度の相対的な関係で決まると考え

られる。しかし、破壊の機構は複雑なので、リンクの強度や災害の強度を正確に把握することは困難である。このため、災害によってリンクが破壊するか否かは確率的に決まるとし、不十分な点を補わざるを得ない。このとき、総所要時間は期待値を考えることになる。

リンクの破壊が確率的に決まるとすると、災害時の道路網の状態は一意に定まらない。たとえばリンク数を m とすれば、2^m の破壊パターンがそれぞれの生起確率をもって得られることになる。そして、すべての場合について使用可能なリンクからなるネットワークでの総所要時間を求め、生起確率を乗じて期待値を計算しなければならない。しかし、これでは目的関数の計算に時間がかかるため、最適交通網構成問題を解くことができない。そのため、ふたつの単純化した問題を考えることにする。

ひとつは、ある災害のもとでリンクは破壊するか否かどちらかに決まると仮定する。そして、総所要時間の最小化を目的関数とする問題である。他方は、ある災害のもとでのリンクの破壊は確率的に決まると仮定する。そして、目的地に到達できるトリップ数の期待値の最大化を目的関数とする問題である。

ここでは、リンクの災害に対する強さを決定変数とする。すなわち、リンクの対災害信頼性の水準を離散的な規格として表わすリンクレベルを設ける。リンクレベルが高ければ、災害に対する強さは増すが、建設費も高くなるとする。これにより、問題は０－１変数を扱う最適交通網構成問題でなく、一般の整数計画問題に対応する最適交通網構成問題となる。

ある地域に展開する道路網に自然災害が及ぶとき、近接するリンクには同じような影響を与えるであろう。すなわち、災害の強さはリンクに独立でなく、地域的分布をもつと考えることができる。地域的分布は災害ごとに異なるであろうが、いくつかの代表的なパターンに分類することが可能であると仮定する。そして、これを災害パターンと呼ぶことにする。それぞれのリンクには、災害パターンごとに定まった強さの災害が加わるとするのである。ここでは簡単のために、災害の強さも離散的に表わせるものとし、これを災害強度レベルと呼ぶことにする。

総所要時間最小化問題では、リンクレベルと災害強度レベルを対応させる。あるリンクの災害強度レベルがリンクレベル以上であるとき、そのリンクは破壊するものとする。また到達トリップ数最大化問題では、リンクレベルと災害強度レベルの組み合わせに対し、リンクの破壊確率あるいは非破壊確率が与えられるものとする。

６．４．３　災害時を考慮した総所要時間最小化問題

ここでは、災害強度レベルとリンクレベルの値から、リンクが破壊するか否かのどちらかに決まると仮定する場合を考える。この場合、ある災害パターンのもとでの使用可能なリンクからなる道路網は一意に定まる。そこで、災害時の道路網における総所要時間の最小化を目的関数とする。ただし、簡単のためにリンクの所要時間は交通量によって変化しないとする。

リンクレベルを決めると、災害パターンごとに災害時の道路網が決まる。それぞれの総所要時間を求め、その加重和の最小化を目的関数とする。需要交通量は災害パターンにかかわらず一定とし、建設費の予算制約だけを制約条件とする。これを定式化すると、次のようになる。

問題６．３

$$\min \quad Z=\sum_{h=1}^{s}\{w_h\sum_{i=1}^{n}\sum_{j=1}^{n}q_{ij}t_{ij}(\boldsymbol{x},\theta_h)\} \qquad (6.27)$$

$$\text{s.t.} \quad \sum_{k=1}^{m} c_k(x_k) \leqq C \qquad (6.28)$$

$$L_k \leqq x_k \leqq L^{max} \quad (k=1, 2, \cdots, m) \qquad (6.29)$$

ここに、 Z ：災害時の総所要時間の加重和

w_h：災害パターンhに対する重み

q_{ij}：ノードij間の需要交通量

t_{ij}：ノードij間の最短所要時間で各リンクのリンクレベルと災害強度レベルによって決まる。経路がないときは、ペナルティーの値をとるとする。

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_m\}$

：ネットワークの状態を表わし、x_kはリンクkのリンクレベルである。

$\theta_h=\{\theta_{h_1}, \theta_{h_2}, \cdots, \theta_{h_m}\}$

：災害パターンhにおける災害の強さで、θ_{hk}はリンクkに対する災害強度レベルを表わす。

c_k：リンクkの建設費で、リンクレベルにより異なる。

C：予算、すなわちネットワーク建設費の上限値である。

L_k：リンクkの改良前のリンクレベル

L^{max}：可能な最高のリンクレベル

m：リンク数

n：ノード数

s：災害パターン数

式（6.27）は総所要時間の加重和の最小化を表わす目的関数である。リンクの破壊のために経路が存在しないノード対があれば、所要時間の代わりにペナルティーを加えるものとする。式（6.28）はネットワーク建設費の予算制約を表わす。式（6.29）はリンクレベルの範囲を表わす。

リンクレベルは非負の整数値をとり、とくに$x_k=0$はリンクが存在しないことを表わすものとする。そして、リンクレベルが高いほど災害に対する強さが増すとする。災害強度レベルも非負の整数値で与え、値が大きいほど強い災害を表わすものとする。$\theta_{hk}=0$の場合は災害が加わらないとする。リンクが災害により破壊されて使用不能になるのは、$x_k \leqq \theta_{hk}$のときである。

問題6.3の最適交通網構成問題としての特徴は、リンクのとり得る状態が多数あることと、目的関数において複数のネットワークを考慮しなければならないことである。このどちらの特徴も、**5.4**の交通網の段階建設問題と同じである。しかし、建設費の予算制約が分かれていないため、問題6.3の方が扱いやすい。

6.4.4 総所要時間最小化問題の解法

問題6.3の最適解は、探索木を用いる分枝後退法で求めることができる。その際、分枝を早い段階で打ち切って計算時間を短縮するには、目的関数の下限値の使用が有効である。そこで、まず目的関数の下限値の計算方法を考える。そして、これを用いる解法を提案する。

目的関数の下限値は、最適交通網構成問題の基本問題における Hoang[15]の場合と同様の考え方で求める。Hoang は最大ネットワークからリンクを除いて実行可能解へ達することを考えていた。問題6.3の場合には、すべてのリンクを最高の状態にしておき、それからリンクレベルを下げて

実行可能解に達するようにすることになる。したがって、次の補助問題を考えればよい。

補助問題６．３．１

$$\min \quad F=\sum_{k=1}^{m}\sum_{l=1}^{L_k^1} f_{kl}\, y_{kl} \qquad (6.30)$$

$$\text{s.t.} \quad \sum_{k=1}^{m}\sum_{l=1}^{L_k^1} c_{kl}\, y_{kl} \geqq C^{max}-C \qquad (6.31)$$

$$y_{kl} \geqq y_{k,\,l+1} \quad (k=1,\ 2,\cdots,m,\ l=1,\ 2,\cdots,L_k^1) \qquad (6.32)$$

$$y_{kl}=0 \ \ or \ \ 1 \quad (k=1,\ 2,\cdots,m,\ l=1,\ 2,\cdots,L_k^1) \qquad (6.33)$$

ここに、　F：実行可能にすることによる目的関数Zの増加の下限値

f_{kl}：リンクkのリンクレベルを$(L^{max}-l+1)$から$(L^{max}-l)$に下げることによる目的関数の増加の下限値

y_{kl}：リンクkのリンクレベルを$(L^{max}-l+1)$から$(L^{max}-l)$に下げるとき１、そうでないとき０とする。

$c_{kl}=c_k(L^{max}-l+1)-c_k(L^{max}-l)$

：リンクkのリンクレベルを$(L^{max}-l)$から$(L^{max}-l+1)$にするのに必要な費用

$C^{max}=\sum_{k=1}^{m} c_k(L^{max})$

：すべてのリンクを最高のリンクレベルにするのに必要な費用

$L_k^1=L^{max}-L_k$

：リンクkのリンクレベルを下げられる範囲を表わす。

式（６.30）は、実行可能にすることにより生じる目的関数Zの増加の下限値の最小化を表わす。式（６.31）は式（６.28）と同じ内容を表わす予算制約である。式（６.32）はリンクレベルを下げる順序に関する制約である。

リンクレベルを下げることによる目的関数の増加の下限値は、そのリンクの両端のノード間の所要時間差から求めることができる。リンクの所要時間を一定としているので、式（６.34）となる。

$$f_{kl}=\sum_{h=1}^{s} w_h(q_{pr}+q_{rp})\{t_{pr}(\boldsymbol{x}_k^l,\theta_h)-t_{pr}(\boldsymbol{x}_k^{l+1},\theta_h)\} \qquad (6.34)$$

ここに、　$\boldsymbol{x}_k^l$：リンクkのリンクレベルが$(L^{max}-l)$で、それ以外はリンクレベルがL^{max}のネットワーク

p, r：リンクkの端点

この補助問題は０－１整数計画問題なので、整数制約を緩めた線形計画問題を解く。このとき、基本問題の場合には問題の性質を利用し、簡単に最適解が得られた。しかし、ここでは式（６.32）の制約があるため、シンプレックス法を使わなければならない。それでは計算時間がかかるので、式（６.32）の制約条件を除くことを考える。

補助問題６．３．１を線形計画法に直した問題で式（６.32）がなければ、f_{kl}/c_{kl}の小さい順に実行可能になるまで$y_{kl}=1$としてゆけばよい。そして、実行可能になる最後の変数だけが小

数値となり、残りの変数の値は0である。したがって、式（6 .35）の関係が成り立つならば、式（6 .32）の制約条件を考えずに簡単な解法を使える。

$$\frac{f_{kl}}{c_{kl}} < \frac{f_{k,\,l+1}}{c_{k,\,l+1}} \qquad (k=1,\ 2,\cdots,m\ ,l=1,\ 2,\cdots,L_k^1) \qquad (6\ .35)$$

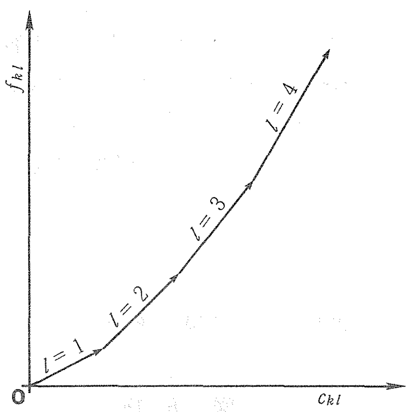
式（6 .35）が成り立たないとき、これを満たさないリンクについては、リンクレベルをいちどに複数段階下げることを1変数で表わすように変換する。すなわち、式（6 .36）のときに、式（6 .37）、（6 .38）の操作を行う。

$$\frac{f_{kl}}{c_{kl}} \geqq \frac{f_{k,\,l+1}}{c_{k,\,l+1}} \qquad (6\ .36)$$

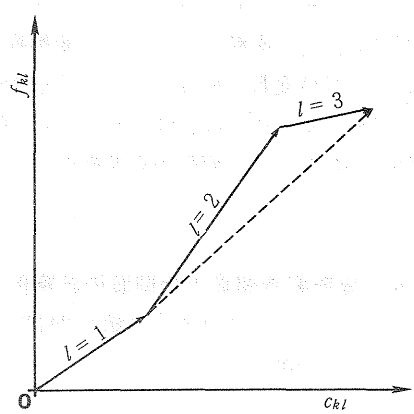
$$f_{kl} \leftarrow f_{kl} + f_{k,\,l+1} \qquad (6\ .37)$$

$$c_{kl} \leftarrow c_{kl} + c_{k,\,l+1} \qquad (6\ .38)$$

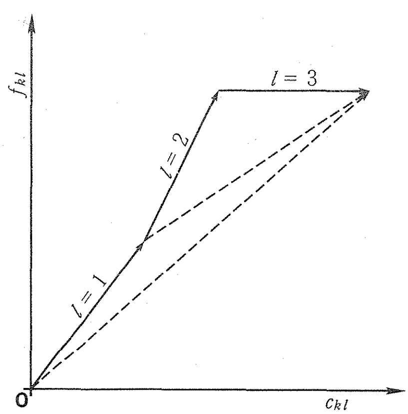
そして、変数の添字を整理し直す。これを繰り返し、式（6 .36）を満たすものがないようにすれば、基本問題の場合と同様にして解くことができる。この変換操作の例を図－6 .12～6 .14に示



図— 6.12　変換操作例 1



図— 6.13　変換操作例 2



図— 6.14　変換操作例 3

す。図－6 .12の場合は式（6 .36）が成り立たないので変換操作は必要ない。図－6 .13、6 .14ではそれぞれ変換操作を行っている。これらの図では、ベクトルで f_{kl}/c_{kl} の傾きを表わし、変換操作後を破線で区別している。

このように変換操作を行った後の線形計画問題は、次の補助問題6．3．2になる。

補助問題6．3．2

$$\min \quad F' = \sum_{k=1}^{m} \sum_{l=1}^{L_k^2} f_{kl}\ y_{kl} \qquad (6\ .39)$$

$$\text{s.t.} \quad \sum_{k=1}^{m} \sum_{l=1}^{L_k^2} c_{kl}\ y_{kl} \geqq C^{max} - C \qquad (6\ .40)$$

$$0 \leqq y_{kl} \leqq 1$$

$$(k=1,\ 2,\cdots,m,l=1,\ 2,\cdots,L_k^2) \qquad (6\ .41)$$

ここに、 L^2_k：変換操作後のリンク k のリンクレベルを下げられる範囲を表わす。

明らかに、$F' \leqq F$ である。よって、目的関数の下限値を式（6 .42）で求めることができる。

$$Z^l = Z^0 + F' \tag{6.42}$$

ここに、 Z^l：目的関数の下限値

Z^0：すべてのリンクレベルを L^{max} としたときの目的関数値

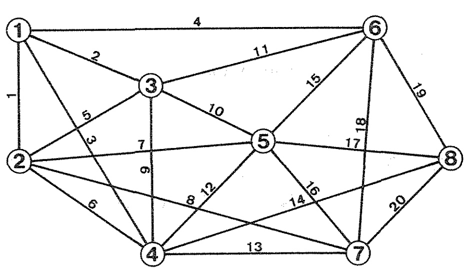
問題6 .3では、ペナルティーを課することにより、解が非連結網であることを許している。そこで、解法には実行可能化分枝だけを用いるアルゴリズムを採用する。これは、補助問題の解を利用し、実行可能解までいちどに分枝するものである。ここでの問題への適用に際し、いくらかの変更が必要である。

問題6 .3での決定変数は x_k のリンクレベルである。しかし、補助問題の解を利用することと、アルゴリズムでの探索木の構造に合わせることのために y_{kl} を分枝変数とする。最初はすべての y_{kl} を自由変数とし、$y_{kl} \leftarrow 1$ として分枝することにより、リンクレベルを下げてゆく。そして、あともどりして $y_{kl} \leftarrow 0$ とすれば、リンク k のリンクレベルはそれよりも下げないことになる。

補助問題6 . 3 . 2を解くときには、変換操作によって一部のリンクの y_{kl} の表わす内容が変わっている。分枝変数に使うときには、元の変数に直しておかなければならない。そして、解を利用して分枝するときには、同一リンクでは順番にリンクレベルが下がるようにする。また補助問題の解で小数値となる変数が変換操作でまとめられたものであるとき、実行可能になればそれ以上分枝しないようにする。

6 . 4 . 5 総所要時間最小化問題の計算例

ここでは、災害時の道路網での総所要時間を最小化する問題6 . 3の例題を解き、どのような解が得られるかを調べる。



図— 6.15 例題 6.1 の最大ネットワーク

例題6 . 1として、図－6 .15のネットワークを用いる。リンクの所要時間や建設費などは、表－6 . 8に示す。リンクレベルの取り得る範囲は、すべてのリンクで0～5とする。需要交通量はリンクでの往復交通量を区別しなくてよいので、表－6 . 9のように三角OD表で与える。この需要交通量を最大ネットワークに最短経路配分すると、リンク交通量は図－6 .16のようになる。災害強度レベルは0～5とする。そして、災害パターンは、図－6 .17に示す4種類とする。災害パターン1は災害の加わらない平常時である。

災害パターンに対する重みは、それぞれ0.60、0.15、0.15、0.10とする。予算は、すべてのリンクを最高のリンクレベルにするのに必要な2798に対し、1100、1500、1900の3種類を考える。その計算の結果、図－6 .18～6 .20の最適解が得られた。これらの最適解は、いずれも平常時には最大ネットワークそのものである。そして、図－6 .16でリンク交通量の多い重要なリンクで、強い災害が加わるもののリンクレベルが高いことがわかる。

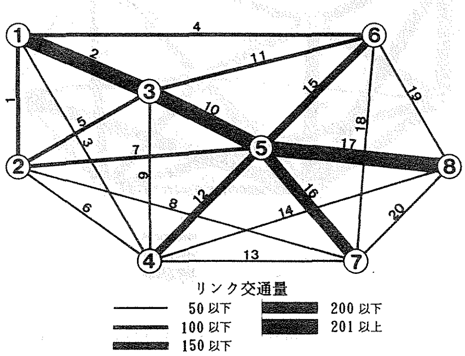
ここでは簡単な例題を解いただけであるが、問題6 . 3の定式化により必要とする道路網案が得られた。すなわち、災害時における総所要時間の短い道路網である。

表— 6.8　例題 6.1 のリンクの所要時間と建設費

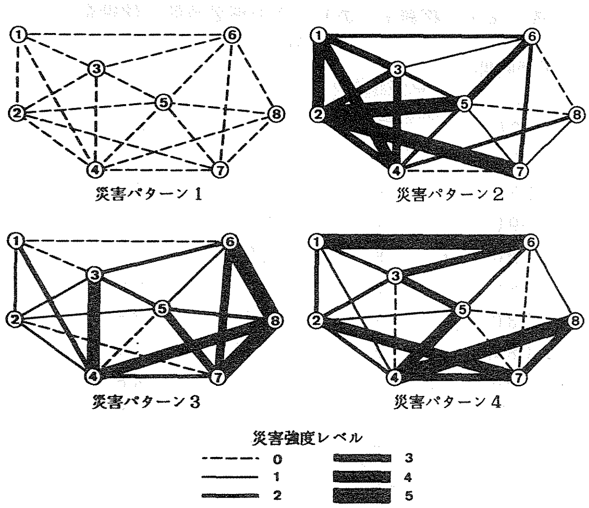
リンク	端	点	所要時間	建	設		費	
				レベル 1	レベル 2	レベル 3	レベル 4	レベル 5
1	1	2	70	9	19	36	53	81
2	1	3	76	15	35	58	85	113
3	1	4	139	4	26	48	80	121
4	1	6	190	7	57	114	192	282
5	2	3	80	9	25	48	71	96
6	2	4	86	16	40	65	90	126
7	2	5	130	27	74	128	188	255
8	2	7	187	13	57	111	179	261
9	3	4	90	5	24	48	86	124
10	3	5	67	5	14	28	49	71
11	3	6	124	15	39	72	112	174
12	4	5	85	18	38	65	103	144
13	4	7	110	1	9	44	91	1 1
14	4	8	168	4	23	49	109	169
15	5	6	85	18	38	61	92	128
16	5	7	78	4	9	14	44	81
17	5	8	100	23	51	87	124	162
18	6	7	120	7	23	55	91	133
19	6	8	80	13	27	42	61	80
20	7	8	71	4	11	24	38	56

表— 6.9　例題 6.1 の需要交通量

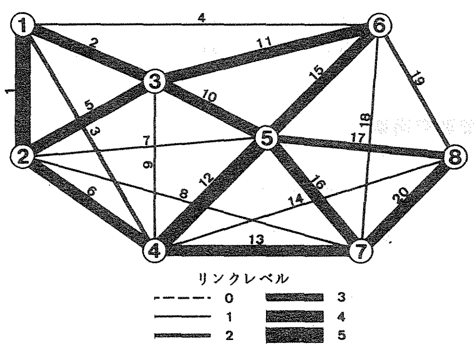
発生	集				中			
	1	2	3	4	5	6	7	8
1		70	97	31	30	98	0	74
2			63	48	20	13	12	57
3				25	10	78	99	38
4					45	75	44	37
5						59	94	63
6							26	9
7								28
8								



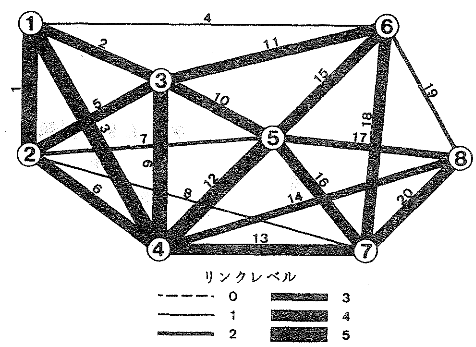
図— 6.16　例題 6.1 の最大ネットワークへの配分結果



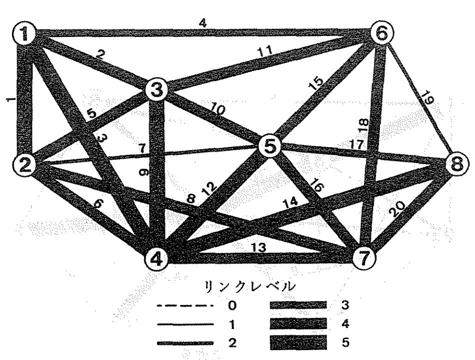
図― 6.17　例題 6. 1 の災害パターン



図― 6.18　例題 6. 1 の予算1100の最適解



図― 6.19　例題 6. 1 の予算1500の最適解



図― 6.20　例題 6. 1 の予算1900の最適解

6.4.6　災害時における到達トリップ数の最大化問題

ここでは、災害時のリンクの破壊は確率的に決まると仮定する場合を考える。この場合には、災害で破壊されるリンクの組み合わせは多数ある。すなわち、使用可能なリンクからなる道路網が、それぞれの生起確率で生じるのである。すべての道路網での総所要時間を計算し、その期待値の最小化を目的関数とすることは不可能といってよい。そこで、目的地に到達できるトリップ数の期待値の最大化を目的関数とする。

決定変数はリンクレベルとし、リンクレベルと災害強度レベルの組み合わせに対してリンクの破壊確率が与えられるとする。そうすると、道路網を決めれば、ある災害パターンにおける個々のリンクの破壊確率が定まる。そのときのノード間に経路が存在する確率を求めれば、目的地に到達可能なトリップ数の期待値が計算できる。このノード間に経路が存在する確率をノード間の信頼度とよぶことにする。

ノード間の信頼度を求めるには、ノード間のあらゆる経路を考えなくてはならない。それでは計算時間がかかるため、最適交通網構成問題の目的関数に使えない。よって、ここではノード間の信頼度そのものでなく、容易に求められる信頼度の下限値を用いる。これを使えば、目的地に到達できるトリップ数の下限値の期待値の最大化を目的関数にすることになる。

問題6.3と同じように、建設費の予算制約だけを制約条件とすると、次のように定式化できる。

問題6.4

$$\max \quad Z=\sum_{h=1}^{s}\{w_h\sum_{i=1}^{n}\sum_{j=1}^{n}q_{ij}\,r_{ij}(\boldsymbol{x},\theta_h)\} \qquad (6.43)$$

$$\text{s.t.} \quad \sum_{k=1}^{m}c_k(x_k)\leqq C \qquad (6.44)$$

$$L_k\leqq x_k\leqq L^{max} \quad (k=1,\,2,\cdots,m) \qquad (6.45)$$

ここに、　Z：目的地に到達できるトリップ数の下限値の加重和

r_{ij}：ノードij間の信頼度の下限値

式(6.43)の目的関数で使用するノード間の信頼度の下限値は、式(6.46)で求めることができる。[16]

$$r_{ij}=\prod_{all\ v\in V_{ij}}\left\{1-\prod_{k\in\Gamma_v}p_k(x_k,\theta_{hk})\right\} \qquad (6.46)$$

ここに、　V_{ij}：ノードijを分けるすべてのカットからなる集合

Γ_v：カットvを構成するリンクの集合

p_k：リンクkの破壊確率

式(6.43)の目的関数は、ノード間に経路が存在する確率の下限値をトリップ数で加重和している。つまり、トリップ数に比例する割合でノード間の信頼度が考慮されるのである。そのため、トリップの少ないノード間の信頼度が極端に低くなることも考えられる。トリップが少ない場合でも、ノード間にはある程度の信頼度を確保したいならば、問題6.4では不十分である。

それぞれのノード間の信頼度を一定値以上にするには、問題6.4に制約式を追加すればよい。しかし、それでは問題が複雑になる。そこで、簡単な方法として信頼度の下限値を対数変換する。

そうすれば、信頼度が極端に低くなるのが避けられる。ただし、目的関数の表わす内容の具体性は失われる。目的関数に信頼度の下限値の対数を用いる問題は、次のようになる。

問題6．5

$$\max \quad Z=\sum_{h=1}^{s}\{w_h \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}\cdot log\, r_{ij}(\boldsymbol{x},\theta_h)\} \qquad (6.47)$$

$$\text{s.t.} \quad \sum_{k=1}^{m} c_k(x_k) \leqq C \qquad (6.48)$$

$$L_k \leqq x_k \leqq L^{max} \quad (k=1,\ 2,\cdots,m) \qquad (6.49)$$

6．4．7　到達トリップ数最大化問題の解法

ここでは、問題6．4あるいは問題6．5として定式化した到達トリップ数最大化問題の解法を示す。これらの問題は目的関数が異なるが、解法の基本的な考え方は総所要時間最小化問題の場合と同じである。探索木を使って実行可能解に達するまでリンクレベルを下げてゆき、最適解を探せばよい。このとき、問題6．3の解法では目的関数の下限値を使用し、分枝を早い段階で打ち切るようにしていた。問題6．4、6．5の場合に、これに相当する目的関数の上限値をどのように求めるかを考えなくてはならない。

目的関数の上限値を求めるには、各リンクのリンクレベルを下げたときの目的関数の減少の下限値が必要である。この下限値が強力な問題6．5について解法を示すことにする。

目的関数の上限値を求めるための補助問題は、次に定式化するものである。

補助問題6．5．1

$$\min \quad G=\sum_{k=1}^{m}\sum_{l=1}^{L_k^1} g_{kl}\ y_{kl} \qquad (6.50)$$

$$\text{s.t.} \quad \sum_{k=1}^{m}\sum_{l=1}^{L_k^1} c_{kl}\ y_{kl} \geqq C^{max}-C \qquad (6.51)$$

$$y_{kl} \geqq y_{k,\,l+1} \quad (k=1,\ 2,\cdots,m,\ l=1,\ 2,\cdots,L_k^1) \qquad (6.52)$$

$$y_{kl} = 0 \ or\ 1 \ (k=1,\ 2,\cdots,m,\ l=1,\ 2,\cdots,L_k^1) \qquad (6.53)$$

ここに、　G：実行可能にすることによる目的関数Zの減少の下限値

g_{kl}：リンクkのリンクレベルを$(L^{max}-l+1)$から$(L^{max}-l)$に下げることによる目的関数の減少の下限値

問題6．5は最大化問題なので目的関数の減少の下限値を求めるが、補助問題6．5．1は補助問題6．3．1と同じものである。したがって、リンクレベルの変換操作も同じように行う。そして、線形計画問題に直した補助問題の解の目的関数値をG'とすると、目的関数の上限値を式(6.54)で求めることができる。

$$Z^u = Z^0 - G' \qquad (6.54)$$

ここに、　Z^u：目的関数の上限値

リンクレベルを下げることによる目的関数の減少の下限値は、他のリンクのレベルを最高の状態にしたネットワークで求めることができる。対象とするリンクのリンクレベルを下げたときの

目的関数の減少値を計算すれば、それが減少の下限値となる。すなわち、式（6.55）である。

$$g_{kl}=\sum_{h=1}^{s}[w_h\sum_{i=1}^{n}\sum_{j=1}^{n}q_{ij}\{log\,r_{ij}(\boldsymbol{x}_k^l,\theta_h)-log\,r_{ij}(\boldsymbol{x}_k^{l+1},\theta_h)\}] \quad (6.55)$$

式（6.55）で目的関数の減少の下限値が求まることは、ノード間の信頼度の下限値の定義式をもとにして証明できる。信頼度の下限値の対数は、式（6.56）である。

$$log\,r_{ij}=\sum_{all\,v\in V_{ij}}log\{1-\prod_{k\in\Gamma_v}p_k(x_k,\theta_{hk})\} \quad (6.56)$$

式（6.56）では、カットの非破壊確率の対数の和を求めていることになる。カットに含まれないリンクのリンクレベルを下げても、カットの非破壊確率は変わらない。そこで、リンクkのリンクレベルを下げたとき、リンクkを含むカットでの非破壊確率の対数の変化を調べる。

いま、ある解$\boldsymbol{x}_1$があり、それからリンクkのリンクレベルを1段階下げたものを$\boldsymbol{x}_2$とする。解$\boldsymbol{x}_3$はリンクkのリンクレベルが$\boldsymbol{x}_1$と同じであり、他のリンクのリンクレベルは$\boldsymbol{x}_1$でのリンクレベル以下であるとする。解$\boldsymbol{x}_4$は、$\boldsymbol{x}_3$からリンクkのリンクレベルを1段階下げたものである。リンクkを含むカットvの非破壊確率を次のように表わす。

$$r_v^1=1-P_v\,p_k \quad (6.57)$$

$$r_v^2=1-P_v\,p_k' \quad (6.58)$$

$$r_v^3=1-P_v'\,p_k \quad (6.59)$$

$$r_v^4=1-P_v'\,p_k' \quad (6.60)$$

ここに、　r_v^i：解$\boldsymbol{x}_i$におけるカットvの非破壊確率

P_v：解$\boldsymbol{x}_1$、$\boldsymbol{x}_2$における、カットvに含まれるリンクk以外のリンクの破壊確率の積

P_v'：解$\boldsymbol{x}_3$、$\boldsymbol{x}_4$における、カットvに含まれるリンクk以外のリンクの破壊確率の積

p_k：解$\boldsymbol{x}_1$、$\boldsymbol{x}_2$におけるリンクkの破壊確率

p_k'：解$\boldsymbol{x}_3$、$\boldsymbol{x}_4$におけるリンクkの破壊確率

リンクkの破壊確率はリンクレベルと災害強度レベルで決まるが、簡単に表わした。P_vとP_v'、p_kとp_k'には次の関係がある。

$$0\leqq P_v\leqq P_v'\leqq 1 \quad (6.61)$$

$$0\leqq p_k\leqq p_k'\leqq 1 \quad (6.62)$$

$\boldsymbol{x}_1$と$\boldsymbol{x}_3$から、それぞれリンクkのリンクレベルを下げたときのカットvの非破壊確率の対数の減少の差は、次のようになる。

$$\begin{aligned}&(log\,r_v^1-log\,r_v^2)-(log\,r_v^3-log\,r_v^4)\\&=log(1-P_v p_k)-log(1-P_v p_k')-log(1-P_v' p_k)+log(1-P_v' p_k')\\&=log\frac{1-(P_v p_k+P_v' p_k')+P_v P_v' p_k p_k'}{1-(P_v' p_k+P_v p_k')+P_v P_v' p_k p_k'}\end{aligned}$$

ここで、

$$(P_v p_k + P'_v p'_k) - (P'_v p_k + P_v p'_k)$$
$$= (P'_v - P_v)(p'_k - p_k) \geqq 0$$

よって、

$$(\log r_v^1 - \log r_v^2) - (\log r_v^3 - \log r_v^4) \leqq 0 \qquad (6.63)$$

式（6.63）は、$\boldsymbol{x}_1$ からリンク k のリンクレベルを下げたときよりも、$\boldsymbol{x}_3$ から下げたときの方がカット v の非破壊確率の対数の減少量が大きいことを示している。したがって、式（6.55）で目的関数の減少の下限値が求まることは明らかである。

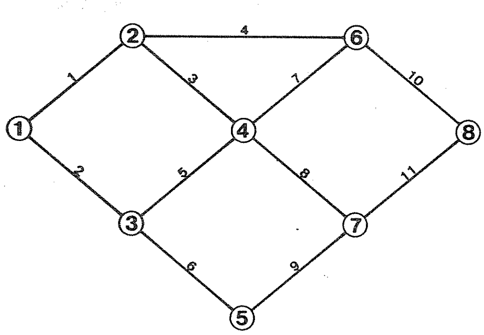
問題6．3では、リンクレベルを下げることによる目的関数の増加の下限値は、リンクの両端のノード間の総所要時間差であった。式（6.56）ではネットワーク全体での目的関数値の差を求めており、問題6．5の場合の方が強力である。

問題6．5では、非連結網が最適解になることはない。そのため、非連結網になる可能性があるならば連結網化分枝を行うアルゴリズムを用いる必要がある。そうでなければ、問題6．3と同じ実行可能化分枝だけを用いるアルゴリズムでよい。問題6．5は最大化問題なので目的関数値の比較の部分が異なるが、計算手順は同じである。

6．4．8　到達トリップ数最大化問題の計算例

ここでは、問題6．5の計算例を示す。問題6．5ではノード間の信頼度の下限値を対数変換しているため、必ずしも到達トリップ数を最大化することにはならない。しかし、ノード間の信頼度が極端に低くならないようにしたうえで、トリップの多いノード間には災害時にも経路が確保される解が得られるはずである。これを計算例で確める。

例題6．2として図－6．21のネットワークを用いる。リンクの建設費などは、表－6．10に示す。問題6．5ではリンクの長さは必要ないが、需要交通量を配分するために設ける。リンクレベルのとり得る範囲は、すべてのリンクにおいて1～4とする。需要交通量は表－6．11に示すもので、最大ネットワークに最短経路配分すると、リンク交通量は図－6．22のようになる。災害強度レベルは0～5とし、リンクレベルと災害強度レベルの組み合わせに対して、リンクの破壊確率を表－6．12のように定める。そして、災害パターンは、図－6．23に示す5種類とする。



図— 6.21　例題 6.2 の最大ネットワーク

災害パターンに対する重みは、それぞれ、0.30、0.30、0.15、0.15、0.10とする。予算は、すべてのリンクを最高のリンクレベルにするのに必要な653に対し、400、450、500の3種類を考える。計算の結果、図－6．24～6．26の最適解が得られた。これらの図を見れば、強い災害を受けないリンクでノード間の経路を確保する解が選ばれていることがわかる。

リンクの破壊確率や建設費が変われば、この計算例とは異なる傾向の解が得られるかもしれな

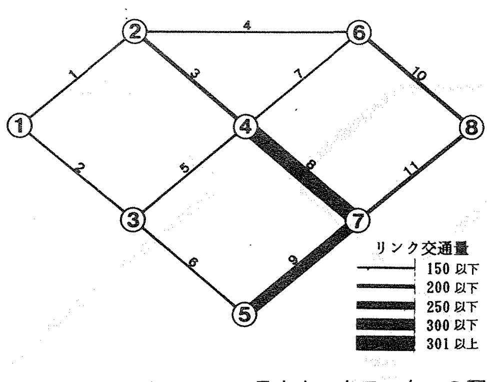
い。しかし、災害時にもノード間に経路を確保するために、問題６．５の定式化は有効であるといえる。

表― 6.10　例題 6.2 のリンクの長さと建設費

リンク	端	点	長さ	建 レベル２	設 レベル３	費 レベル４
1	1	2	11	8	26	51
2	1	3	14	11	30	57
3	2	4	16	13	36	68
4	2	6	23	19	50	93
5	3	4	18	17	45	84
6	3	5	15	13	34	65
7	4	6	17	14	39	75
8	4	7	9	6	21	42
9	5	7	10	7	24	48
10	6	8	8	5	18	37
11	7	8	7	4	16	33

表― 6.11　例題 6.2 の需要交通量

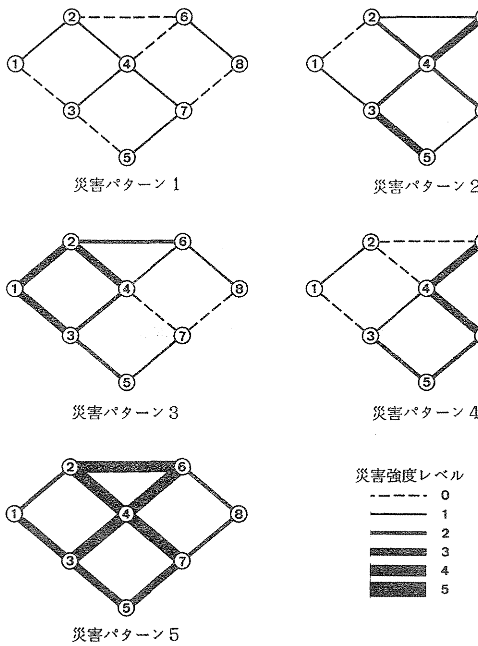
発生 \ 集中	1	2	3	4	5	6	7	8
1		34	51	0	73	25	19	42
2			21	38	64	17	53	27
3				14	9	92	0	37
4					59	16	81	46
5						34	76	14
6							42	17
7								23
8								



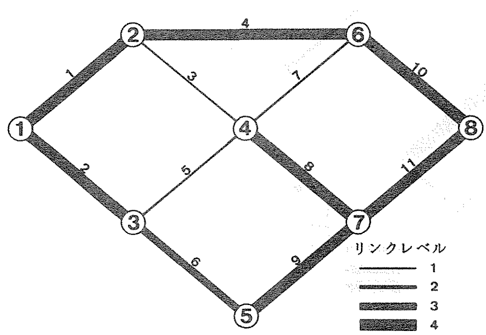
図― 6.22　例題 6.2 の最大ネットワークへの配分結果

表— 6.12　例題 6. 2 のリンクの破壊確率

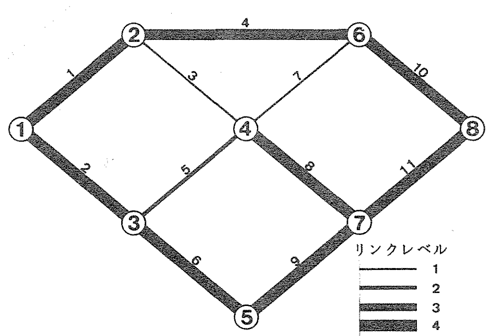
リンクレベル	破壊確率 災害強度 0	災害強度 1	災害強度 2	災害強度 3	災害強度 4
1	0.00	0.70	0.90	0.92	0.94
2	0.00	0.50	0.70	0.90	0.92
3	0.00	0.30	0.50	0.70	0.90
4	0.00	0.10	0.30	0.50	0.70



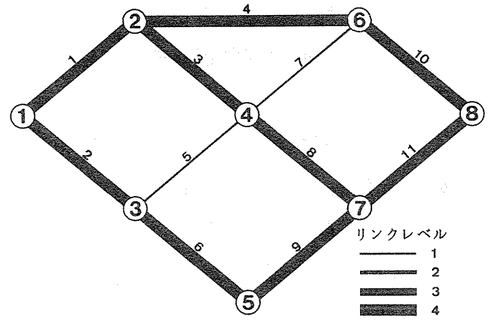
図— 6.23　例題 6. 2 の災害パターン



図— 6.24　例題 6. 2 の予算400の最適解



図— 6.25　例題 6. 2 の予算450の最適解



図— 6.26　例題 6.2 の予算500の最適解

6.4.9　むすび

ここでは、災害に対する信頼性を最大にするように道路網を計画する問題を扱った。道路網の災害に対する信頼性の評価について検討し、最適交通網構成問題として定式化した。そのひとつは、災害時の総所要時間を最小化するもので、もうひとつは災害時の到達トリップ数を最大化する問題である。そして、それぞれの問題に対する解法を示した。

災害時の道路網の状態を表わすために、リンクレベルや災害強度レベル、災害パターンなどの概念を導入した。ここで対象とした問題は、これらが前提となっている。しかし、実際の自然災害を分析して導いたものではないため、その妥当性についてはなお検討を重ねることが必要である。

問題の解法の基本的な部分は、最適交通網構成問題の基本問題に対するものと同じである。決定変数をリンクレベルとしたことへの対応と、目的関数の下限値あるいは上限値の計算が変更部分である。災害の扱い方を変えれば、解法も修正することが必要になる。そして、実用的なものにするならば、近似解法にしなくてはならない。

ここで扱った問題は、実際の自然災害と道路網との関係を十分に表現できていないかもしれない。しかし、従来は考えられていなかった、災害時のネットワークとしての信頼性を考慮して道路網計画を行う方法を示したことに意義がある。

6.5　配分対象道路網作成問題への適用[17)]

6.5.1　はじめに

道路区間の交通量や走行速度を推定し、地域の交通流動の状況を知るために、分布交通量を道路網へ配分する作業が行われる。このとき、実際の道路網そのものが用いられることはほとんどなく、適当に集約した仮想の配分対象道路網が設定される。それは、細街路の状態を知る必要がない場合が多く、ネットワークの規模が大きければ、交通配分に要する計算量が膨大になるからである。

配分対象道路網の設定によく使われる方法は、主要幹線から順次採用するなどして道路網案をつくり、配分と道路網案の修正を繰り返すものである。この作業は一般に極めて恣意性の高いも

のであり、手作業と多量の繰り返し計算が必要である。そして、配分結果は設定される道路網に応じて変化する。したがって、信頼できる配分結果を得るためには、想定する道路網の設定について十分に検討しなければならない。また、配分対象道路網を容易に決めることができれば、むだな計算や作業を省き、交通配分を迅速に行えるようになる。

これまで、配分手法に関しては多くの研究がなされているが、配分対象道路網の設定についての研究は少ない。道路網の設定と配分手法は、どちらも交通配分作業に欠くことのできないものである。道路網を適当に決めるのであれば、配分手法に凝っても無意味である。従来の交通配分に関する研究は配分手法に片寄りすぎていたといえる。

配分対象道路網に関するものでは、ゾーンとネットワークのアグリゲーション問題の研究[18)]がある。しかし、問題の本質を十分に解明したものでなく、実用性にも遠い。そこで、ここではどのような配分対象道路網をつくればよいかを検討し、それに基づいて道路網案を客観的な基準で作成する手法を導く。

6.5.2では、配分の対象とする道路網は、どのような条件を満たすものでなければならないかを検討する。

6.5.3では、配分対象道路網作成問題の前提条件を示し、配分対象道路網の作成手順について述べる。

6.5.4では、配分の対象とするリンクを選び出す問題を最適交通網構成問題として定式化する。

6.5.5では、最適交通網構成問題の近似解法を応用した配分対象リンク選択問題の解法を提案する。

6.5.6では、配分対象リンク選択問題の解のネットワークにおける、配分結果に影響しないノードとリンクの集約について述べる。

6.5.7では、作成した配分対象道路網を構成するリンクの交通容量を決める方法を示す。

6.5.8では、提案した手法による配分対象道路網作成の計算例を示す。

6.5.2　配分対象道路網についての検討

交通配分を行う目的はさまざまであり、配分結果から得たい情報も場合により異なる。したがって、どのような道路網が配分対象として適しているかも、場合により異なってくる。そこで、ここではある区域または路線に属する道路区間の交通量と走行速度を推定する場合を考える。

実際の道路網を用いないで配分対象道路網を設定する目的は、ネットワークを簡略化して交通配分に要する計算量を減らすことである。そのため、配分対象道路網は必要とする程度に集約されたものでなければならない。これが必須の条件である。そして、配分に用いる分布交通量はゾーン分割を行って求められているので、道路網もこれと整合していなければならない。

道路網での配分交通量を推定するには等時間配分が使われる。このとき、現況の分布交通量を配分した結果が路側観測されている区間交通量と一致すれば、よい配分対象道路網が得られたとするのが一般的である。ここでも、この評価方法を踏襲する。しかし、配分結果を観測値に一致させるとしても、つねにすべての道路区間での観測値が得られているわけではない。さらに、計

画あるいは関心の対象の道路区間以外では、観測値と一致する配分結果を必ずしも与えなくてよい。それは、特定の道路区間の交通量を推定することが交通配分の目的であり、その他の情報を要求していないからである。これらのことから、少なくとも定められた道路区間では配分結果が観測値と一致するか、一致すると考えられる道路網であればよい。

配分対象道路網は現況の分布交通量を用いて作成する。しかし、交通量を推定するときには、これとは異なる分布交通量を配分することになる。この場合においても、特定の道路区間では配分結果が観測されるであろう区間交通量と一致しなければならない。現況の資料だけを用いて道路網を求めるときに、この条件を満たすのは容易でない。しかし、道路網全体として見たときに、実際の道路網と配分対象道路網とで交通の流れが大きく変わらなければ、分布交通量が異なっても配分交通量の精度が保てると考えられる。すなわち、地域の交通流動のパターンが変わらない道路網であることが必要である。

以上をまとめると、次の条件を満たす配分対象道路網をつくればよい。

① 必要最小限の構成要素だけでネットワークがつくられていること。

② 定められた道路区間の交通量と走行速度が観測値と一致すること。

③ 地域の交通流動のパターンが、実際の道路網の場合と大きく変わらないこと。

6．5．3　配分対象道路網の作成手順

まず、ここで扱う配分対象道路網作成問題を明確にしておく。ネットワークは道路区間をリンクとし、その端点をノードとして構成する。それらのノードのいくらかが、トリップの起終点になる発生・集中ノードであるとする。与えられるデータは、対象地域内で検討しなければならないすべてのリンクの長さ、交通容量、接続関係、交通量と走行速度または所要時間との関係式、それに分布交通量であるとする。現況の分布交通量と対応してリンクの交通量や走行速度などが調査されている場合、それらの観測値も使用できるものとする。

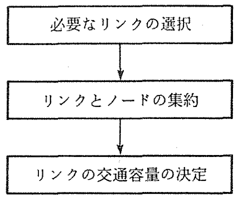
求めるのは、交通配分に用いる道路網である。すなわち、リンクとノードの接続関係と交通容量などのリンク特性である。発生・集中ノードをまとめる場合には、分布交通量の集約方法も求めなければならない。

従来の配分対象道路網を作成する作業では、代表的なリンクを選んで複数のリンクをまとめ、新たな位置に仮想のリンクを設けて道路網をつくっていた。そのうえで、交通配分とリンクの長さや交通容量、接続関係の調整とを繰り返し、期待する配分結果が得られるようにしていた。しかし、仮想のリンクを設ける方法は数理計画問題として表わすことが容易でない。そこで、与えられた詳細な道路網からリンクを除くことによって配分対象道路網をつくることにする。すなわち、詳細な道路網を最大ネットワークとし、最適交通網構成手法を適用して配分の対象とするリンクを選択する。

このようにして求めた道路網は、そのまま配分計算に使える。しかし、リンクを除いたことにより、これまで複数のリンクとして表わしていたものが、1本のリンクで表わせるような場合がある。そこで、配分結果に影響を与えないリンクとノードの集約を行う。そして最後に、定めた道路網を構成するリンクの交通容量を決める。これは、除かれたリンクが分担していた交通容量

の影響を調整するためである。さらに、リンクの交通容量を決め直すことにより、トリップの起終点をいくつかのノードにまとめることによる影響を打ち消す効果も期待できる。

以上の配分対象道路網の作成手順をまとめたのが図－６.27である。



図― 6.27　配分対象道路網の作成手順

６.５.４　配分対象リンク選択問題

詳細なネットワークから配分の対象とするリンクを選び出すのが、配分対象道路網作成の主要部分である。この問題は最適交通網構成問題として扱うことができる。その定式化に際しては、得られる解が６.５.２で述べた条件を満たすようにしなければならない。そこで、これらの条件を考慮し、次のように目的関数と制約条件を決める。

いくらかのリンクでは路側で交通量が観測されており、これと配分結果が一致するよう求めらている。また交通量は観測されていないが、そのリンクの交通量を検討の対象とする場合もある。ここでは、これらの特定のリンクを着目リンクとよぶことにする。着目リンクの交通量と走行速度を観測値に一致させるのは、リンクの交通容量の修正で行う。これについてはあとで述べるが、できるだけ小さな修正ですむようにしておかなくてはならない。そのためには、着目リンクの交通量に大きな影響を与える起終点間のトリップが、配分対象道路網においても元の道路網の場合と同じ経路をとることが望ましい。そこで、ネットワークの集約によって、各起終点間のトリップが着目リンクの交通量に及ぼす影響を調べる。そして、大きな影響を与える可能性のある起終点間のトリップが、集約した解のネットワークにおいても詳細なネットワークと同じ経路をとるように、目的関数で重みを与えることにする。

つぎに、地域の交通流動のパターンを大きく変えないようにするという点について考える。交通流動のパターンは、交通量の大きな起終点間のトリップがどのような経路をとるかでほぼ決まる。リンクの交通容量が需要交通量より極端に小さくなければ、等時間配分においても交通量の大きな起終点間のトリップの大部分は、最短経路配分の場合と同じ経路をとると考えられる。よって、総走行時間あるいは総走行距離が最小になるようなネットワークを選べば、交通量の大きな起終点間のトリップの経路は変化しないと考えられる。すなわち、交通流動のパターンが保持されるのである。

リンクの所要時間を交通量によって変化させるのは、大規模な最適交通網構成問題を解くときには困難である。ここではリンクの所要時間を一定とし、総走行時間の最小化を行うことにする。そして、必要最小限の構成要素からなるネットワークとするため、リンクの数をできるだけ減ら

すことにする。しかし、リンクを減らすことと総走行時間を最小化することは、同時に実現できない。そこで、リンク数は制約式として問題に組み込み、リンク数制約の下で総走行時間を最小化する。

これらをまとめれば、配分対象リンク選択問題は、リンク数制約下で重みつき総走行時間を最小化する最適交通網構成問題になる。定式化すると、次のようになる。

問題6.6

$$\min \quad Z=\sum_i \sum_j w_{ij} q_{ij} t_{ij}(\boldsymbol{x}) \tag{6.64}$$

$$\text{s.t.} \quad \sum_k x_k \leqq M \tag{6.65}$$

$$x_k = 0 \ or \ 1 \tag{6.66}$$

ここに、 Z ：走行時間の加重和

w_{ij}：ノードij間のトリップに対する重みで、着目リンクへの影響の大きさを表わす。

q_{ij} ：ノードij間のトリップ数

t_{ij} ：ノードij間の最短所要時間で、ネットワークの状態により異なる。

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_k, \cdots\}$

：ネットワークの状態を表わし、リンクkがネットワークに含まれるとき$x_k=1$、含まれないとき$x_k=0$とする。

M ：ネットワークに含まれるリンク数の制約値

式（6.64）は、着目リンクへの影響の大きさを表わす重みを使って走行時間を加重和し、それを最小化する目的関数である。式（6.65）は、ネットワークに含まれるリンク数を一定値以下とする制約式である。この問題6.6の構造は、最適交通網構成問題の基本問題とよく似ている。

問題6.6の特徴は、トリップによって目的関数での重みが異なることである。この重みの決め方が、解のネットワークが配分対象道路網として適するものかどうかを左右する。ネットワークを集約したときに、ある起終点間のトリップの経路の変化が着目リンクに関係する可能性の大きさを表わすのがこの重みである。言い換えれば、トリップの経路と着目リンクとの関連の強さを表わすといってもよい。したがって、トリップに対する重みは、ネットワークにおけるトリップの起終点と着目リンクの位置によって決まるものでなければならない。つまり、重みを任意に与えるのでなく、ネットワーク構造によって重みが決まる方法を考えなくてはならない。

最大ネットワークから除くリンクの組み合わせによるトリップの経路の変化をあらかじめ予測し、それに基づいて重みを決めることは実際上不可能である。そのため、直観的な方法で重みを決めるしかない。また作成した道路網を等時間配分に用いる場合、トリップの重みも等時間配分を行った状態で調べるべきである。しかし、最大ネットワークで等時間配分するということは交通配分の解を求めることである。それでは配分対象道路網を作成する意味がない。そこで、等時間配分を用いないで重みを決めなくてはならない。ここでは、次のようにしてトリップに対する重みを決める。

まず、ノードij間のトリップと着目リンクkについて考える。最大ネットワークで最短経路を

求めると、次のどちらかになる。

① リンクkがノードij間の最短経路に含まれる。

② リンクkはノードij間の最短経路に含まれない。

①の場合、最短経路に含まれるリンクのどれかがネットワークから除かれると、ノードij間の最短経路がリンクkを通らなくなる可能性がある。すなわち、ノードij間のトリップが着目リンクkに与える影響は大きいと考えられる。逆に②の場合には、最短経路に含まれるリンクのどれかがネットワークから除かれたとき、ノードij間の最短経路がリンクkを通るようになる可能性がある。最短経路とリンクkを含む経路との距離の差が小さいほど、その可能性が高いと考えられる。

ここでは、リンクの長さは所要時間で考えている。そこで、最短所要時間とリンクkを通る経路の中での最短所要時間との比で、トリップの経路とリンクkとの関連の強さを表わすことにする。これは、式（6.67）のようになる。

$$w_{ij}^{k} = \frac{t_{ij}^{0}}{t_{ij}^{k}} \qquad (6.67)$$

ここに、 w_{ij}^{k}：ノードij間のトリップの経路とリンクkとの関連の強さを表わす指標

t_{ij}^{0}：最大ネットワークでのノードij間の最短所要時間

t_{ij}^{k}：最大ネットワークで、リンクkを通るノードij間の経路の中での最短所要時間

式（6.67）ではリンクkがノードij間の最短経路に含まれるとき$w_{ij}^{k}=1$となり、それ以外のときには$0 \leqq w_{ij}^{k} < 1$となる。

着目リンクが1本だけのときは、w_{ij}^{k}をもってトリップに対する重みとしてよい。しかし、複数の着目リンクがあるのが一般的な場合である。このとき、経路の変化によって同じ交通量が増減しても、それぞれの着目リンクによって影響が異なる。そこで、着目リンク間の相対的な重みはリンクの交通容量に反比例させる。すなわち、w_{ij}^{k}をリンクkの交通容量の逆数で加重和すればよい。この場合、着目リンクを通過する経路が考えられないときは、重みが0になる。しかし、着目リンクへの影響はなくとも、交通流動パターンを表わすには重要なことがある。この点を考慮し、式（6.67）とつぎの式（6.68）～（6.70）でトリップに対する重みを決めることにする。

$$w_{ij}' = \sum_{k} \frac{w_{ij}^{k}}{c_k} \qquad (6.68)$$

$$w_0 = \min_{i,j} \left(w_{ij}' \mid w_{ij}' > 0 \right) \qquad (6.69)$$

$$w_{ij} = \max\left(\frac{w_{ij}'}{w_0},\ 1 \right) \qquad (6.70)$$

ここに、c_k：リンクkの交通容量

式（6.68）は各着目リンクとの関連の強さを加重和することを表わしている。式（6.69）、（6.70）は、それらの中の正の最小値を基準としてトリップに対する重みを決めることを表わしている。そして、式（6.70）では着目リンクへの影響がないと計算されたトリップに対しても、最小の影響を及ぼすトリップと同じ重みを与えている。

6.5.5　配分対象リンク選択問題の解法

問題6.6には、最適交通網構成問題の基本問題に対する解法が応用できる。配分対象道路網を作成しようとするとき、道路網の規模は大きいのが普通である。したがって、大規模問題を扱うことを考慮した近似解法でなければならない。そこで、4.5で提案した簡易 forward 法を基礎に解法を考える。

問題6.6の場合、最適交通網構成問題を解く前に、トリップに対する重みを求めなければならない。式（6.67）～（6.70）に従うと、最大ネットワークでノード間の最短所要時間とそれぞれの着目リンクを通る経路の中での最短所要時間を求めなければならない。着目リンクを通る経路での最短所要時間は k 最短路問題を使って求めることができる。所要時間の短い順に経路を列挙し、着目リンクを通るものを探せばよい。しかし、この方法ではノード対の組み合わせごとに k 最短路問題を解く必要がある。それでは、大規模な問題を扱うのに適さない。ここでは、式（6.71）を用いて近似的な値を求めることにする。

$$t_{ij}^{k}=t_{ip}(\boldsymbol{x}_r)+l_k+t_{rj}(\boldsymbol{x}_p) \qquad (6.71)$$

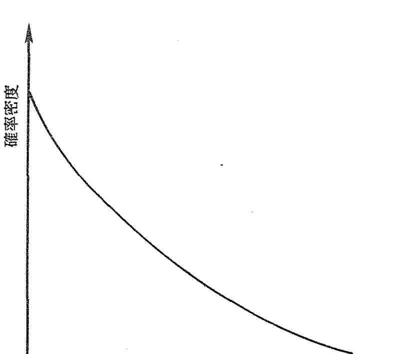
ここに、　$\boldsymbol{x}_r$：最大ネットワークからノード　に接続するリンクを除いたネットワーク

l_k：リンク k の所要時間

p, r：それぞれリンク k の始点と終点のノード

ただし、着目リンク k を有向リンクとして扱っている。着目リンクの両方向を考えるときには、方向別に式（6.71）を適用する。ノードip間とノードrj間の最短所要時間を求めるときには、それぞれノード r、p に接続するリンクを除く。これは、リンク k を通る非現実的な経路を除くためである。このようにしても、t_{ij}^{k}を与える経路がループを含む可能性がある。しかし、そのような経路を除くには、多くの計算が必要である。そして、式（6.71）を使うならば、ノード対ごとに計算をすべてやり直さなくてもよい。リンク k の両端点と各ノード間の最短所要時間を求めるだけである。よって、ループを含む経路があっても、式（6.71）を用いることにする。

この方法でも、重みを決めるためには何回か最短経路問題を解かなくてはならない。大規模な道路網を扱うことが前提なので、最短経路問題を繰り返し解くことは好ましくない。さらに計算を省略する必要がある。一般にトリップ長分布は、図－6.28に示すような形になり、指数分布にほぼ従うことが知られている。そうすると、着目リンクから離れた発生・集中ノード間のトリップは、そのノード間のトリップ全体として及ぼす着目リンクの交通量への影響が小さいと考えられる。そこで、各着目リンクと経路との関連の強さを調べるのは、着目リンクから一定の距離以内にある発生・集中ノード相互間だけとする。



図― 6.28　トリップ長分布

トリップの経路と着目リンク k との関連の強さを表わす指標は、次の手順で求める。ただし、最大ネットワークにおけるノード間の最短所要時間は求められているものとする。また、T は着目リンクとの関連を調

べる範囲を決めるパラメータである。

ステップ1：ノード r に接続するリンクを最大ネットワークから除き、発生・集中ノードからノード p までの最短所要時間を所要時間の短い順に求め、$t_{ip}(\boldsymbol{x}_r)$ とする。ただし、$t_{ip}(\boldsymbol{x}_r) > T$ となったときに経路探索を打ち切る。$t_{ip}(\boldsymbol{x}_r) \leqq T$ であるノード i の集合をノード集合 I とする。

ステップ2：ノード p に接続するリンクを最大ネットワークから除き、ノード r から発生・集中ノードまでの最短所要時間を所要時間の短い順に求め、$t_{rj}(\boldsymbol{x}_p)$ とする。ただし、$t_{rj}(\boldsymbol{x}_p) > T$ となったときに経路探索を打ち切る。$t_{rj}(\boldsymbol{x}_p) \leqq T$ であるノード j の集合をノード集合 J とする。

ステップ3：ノード集合 I に属するノード i と、ノード集合 J に属するノード j の組み合わせをつくり、式（6.67）で w_{ij}^k を計算する。

ステップ4：$i \in I,\ j \in J$ が同時に成り立たないノードの組み合わせに対しては $w_{ij}^k = 0$ とする。

最短経路探索は、ある起点のノードから他のすべてのノードまでの最短経路を求めるDijkstraのアルゴリズム[19]を用いる。ここでは、その計算を途中で中止させるのである。また、ステップ1では普通の場合とは逆に、他のすべてのノードからある終点のノードまでの最短経路を求めなければならない。これは、リンクの始点と終点を逆にすれば、同じ最短経路探索のアルゴリズムで計算できる。

すべての着目リンクに対して w_{ij}^k が求まれば、式（6.68）～（6.70）によって容易に重みを求めることができる。

最適交通網構成問題の解法には簡易 forward 法を用いる。これは、最大ネットワークで最短経路探索を行った結果を使って各リンクの評価値を計算し、forward 法の手順で近似解を求めるものである。問題6.6の場合には、次の式（6.72）、（6.73）でリンクの評価値を決めることにする。

$$f'_k = \sum_i \sum_j \delta_{ijk}\, w_{ij}\, q_{ij} (u_{ij}^0 - t_{ij}^0) \tag{6.72}$$

$$f_k = f'_k + \sum_h (1 - x_h)\, r_{kh}\, f'_h \frac{l_k}{u_{pr}^0} \tag{6.73}$$

ここに、 f'_k：リンク k の最短経路としての評価値

δ_{ijk}：リンク k が最大ネットワークにおけるノード ij 間の最短経路に含まれるとき1、含まれないとき0とする。

u_{ij}^0：最大ネットワークにおけるノード ij 間の第2最短経路の所要時間

f_k：第2最短経路としての代替分を含むリンク k の評価値

r_{kh}：リンク k がリンク h の両端のノード間の第2最短経路に含まれるとき1、含まれないとき0とする。

p, r：リンク h の両端のノード

これらの式は、式（4.8）、（4.9）を目的関数と制約式の違いに合わせて修正したものである。

簡易 forward 法では、最初に生成木をつくるようにリンクを採択していた。それは、非連結

網は目的関数値が無限大になるからである。問題6.6の場合も同じである。しかし、この問題は総走行時間を最小にする道路網を求めるのが本来の目的ではない。着目リンクの交通量や交通流動パターンに対して重要でなければ、あるノードを起終点とするトリップを省いてしまってもかまわない。むしろ積極的に重要でないノードをネットワークから除くことによって、簡略化した配分対象道路網を作成するのに役立つと考えられる。そこで、ネットワークを連結にする手順を省き、単にリンク評価値の大きいものから採択してゆくことにする。

重みの計算の部分も含め、アルゴリズムの概略を示すと、次のようになる。

ステップ1：最大ネットワークにおいて、発生・集中ノード間の最短経路と最短所要時間、第2最短経路の所要時間、それにリンクの両端点間の第2最短経路を求める。

ステップ2：発生・集中ノード間の経路と着目リンクとの関連の強さを調べ、トリップに対する重みを決定する。

ステップ3：各リンクの最短経路としての評価値を求める。

ステップ4：初期解として着目リンクだけからなるネットワークを与える。

ステップ5：そのときのネットワークの状態に対してリンクの評価値を計算する。

ステップ6：まだネットワークに含められていないリンクの中で、評価値が最大のリンクをネットワークに含める。

ステップ7：ネットワークに含まるるリンク数が制約値に達していれば、計算を終える。制約値に達していなければステップ5へもどる。

これで明らかなように、最大ネットワーク全体で最短経路問題を解くのは1回だけである。そのため、対象とする道路網の規模が大きくても、この解法は適用できる。

6.5.6 リンクとノードの集約

配分対象リンク選択問題の解のネットワークは、そのまま配分計算に使ってよい。しかし、修正を加えれば配分計算が容易になることがある。ここでは、いくつかの場合を取り上げ、リンクとノードの集約によってネットワークを整える方法を示す。

ネットワークの特性を変えずにリンクとノードを集約できるのは、次の場合である。

① ノードに接続するリンクがない場合

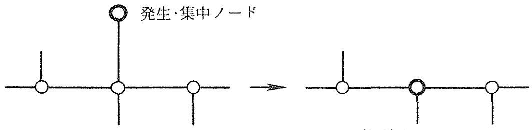
② 発生・集中ノードに接続するリンクが1本だけの場合

③ 発生・集中ノード以外のノードに接続するリンクが1本だけの場合

④ 発生・集中ノード以外のノードに接続するリンクが2本だけの場合

①の場合は、孤立したノードをネットワークから除く。それが発生・集中ノードのときは、そのノードを起終点とするトリップは交通配分に用いない。そのようなトリップを省いても、着目リンクの交通量への影響は小さく、交通流動パターンもあまり変化しないとするのである。

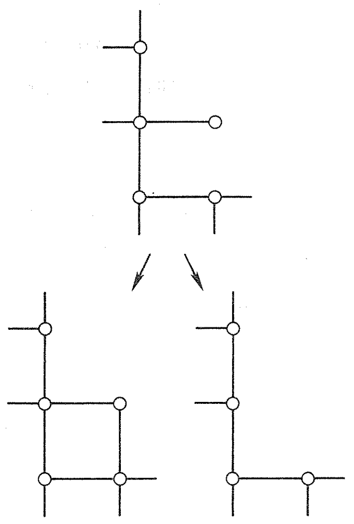
②の場合は、図－6.29に示すように接続しているリンクを除き、発生・集中ノードを他方の位置へ移す。リンクのもう一方の端点が発生・集中ノードのときは、分布交通量をまとめる。除いたリンクの交通量が必要なら



図— 6.29 リンクとノードの集約例1

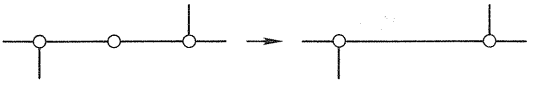
ば、簡単に計算できる。

図－６.30に示す③の場合、そのリンクには交通配分されない。したがって、ノードとリンクを除いても配分結果は同じである。リンクを残したいのなら、リンクに交通が流れるようにしなければ意味がない。このときは、そのノードに接続する他のリンクをネットワークに含める必要がある。このどちらにするかを決めるには、リンク評価値を判断基準としてもよい。除去あるいは付加を検討するリンクの評価値の平均値と、最後にネットワークに付加されたリンクの評価値を比べ、どちらにするかを決めるのである。



図— 6.30　リンクとノードの集約例 2

図－６.31に示す④の場合には、それらのリンクの交通量は等しくなる。そこで、２本のリンクの間のノードを除き、１本のリンクで表わすことができる。交通容量や交通量と所要時間の関係が同じであれば、それらの特性値をそのまま用いればよい。そうでなければ、リンクの所要時間を求めるときには、元のリンクに分けて計算する必要がある。



図— 6.31　リンクとノードの集約例 3

ここに示したのは基本的なものであり、これを組み合わせた場合もある。したがって、これらの場合に相当する部分がなくなるまで、操作を繰り返してノードとリンクを減らしてゆけばよい。

６.５.７　リンクの交通容量の決定方法

作成した配分対象道路網は、与えられた詳細な道路網からリンクやノードを除いたものである。そのため、残されたリンクも元の道路網とは異なる機能を果たすことになる。したがって、与えられた交通容量の値をそのまま使うことはできない。リンクの機能の変化に合わせて修正する必要がある。ここでは、配分対象道路網を構成するリンクの交通容量の決定方法を示す。

配分対象道路網へ交通配分したとき、リンクの交通量が詳細な道路網への配分結果と一致すればよいのである。しかし、リンクを除いているので、特殊な場合以外は交通量は一致しない。そこで、配分対象道路網での配分結果から詳細な道路網での配分結果を推定する。そのために、リンクの交通容量および交通量と所要時間との関係式を定め、式（６.74）を仮定する。

$$g_k(c_k, v_k) = g_k(c_k^*, v_k^*) \qquad (6.74)$$

ここに、g_k：リンク k の所要時間を表わす関数で、交通容量をパラメータ、交通量を説明変数とする。

v_k：詳細な道路網に交通配分したときのリンク k の交通量

v_k^*：配分対象道路網に交通配分したときのリンクkの交通量

c_k^*：配分対象道路網でのリンクkの交通容量

式（6.74）は、ふたつの道路網への配分結果でリンクの所要時間が等しいということである。この仮定を用いるために、あらかじめ配分対象道路網でのリンクの交通容量を決めておく。そして、詳細な道路網での交通量は、配分対象道路網での配分結果から式（6.74）を使って逆算する。

配分対象道路網でのリンクの交通容量を決めるには、ふたつの道路網での配分結果が必要である。この配分結果は、配分対象道路網に適用するのと同じ配分手法によるものであることが望ましい。すなわち、等時間配分によるものであればよい。けれども、重みの決定方法のところで述べたように、詳細な道路網での等時間配分結果は使えない。最短経路配分を用いざるを得ないのである。そこで、配分対象リンク選択問題を解くときに、最大ネットワークでの最短経路探索と同時に交通配分を行っておく。そうすれば、あらためて交通配分し直す必要がなく、計算時間を短縮できる。

着目リンクで交通量や所要時間の観測値が得られている場合、配分結果を一致させなければならない。そのための方法として、配分対象道路網でそれらのリンクの所要時間を固定して等時間配分を行うことにする。そして、そのリンクの交通容量は配分結果から決める。

6.5.8　配分対象道路網作成の計算例

ここでは、提案した方法によって例題を解き、期待する配分対象道路網が得られるかどうかを調べる。例題の道路網と需要交通量は、6.3.5で用いたものを使う。必要とする配分対象道路網が得られたかどうかは、着目リンクの交通量の実測値と推定値の差で調べる。しかし、図－6.9の道路網での交通量の実測値はない。そこで、この道路網へ等時間配分したリンク交通量を実測交通量と見なす。すなわち、配分対象道路網を使った配分結果から推定した着目リンクの交通量が、詳細な道路網での配分結果と一致するかを調べることになる。

着目リンクは、表－6.13に示す3種類とする。この3ケースについて、それぞれリンクを全体の½の26本まで減らした配分対象道路網を作成する。

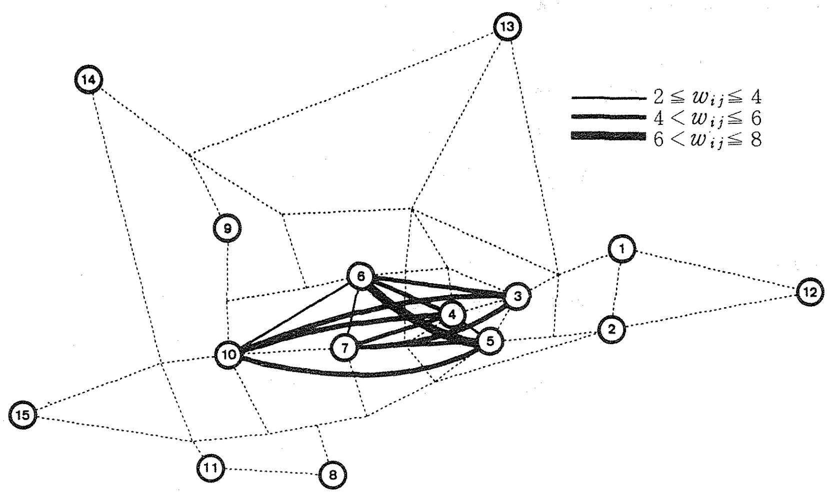
表－6.13　配分対象道路網作成の例題の着目リンク

ケース	着目リンク
1	19
2	3, 7, 9
3	45, 46, 47

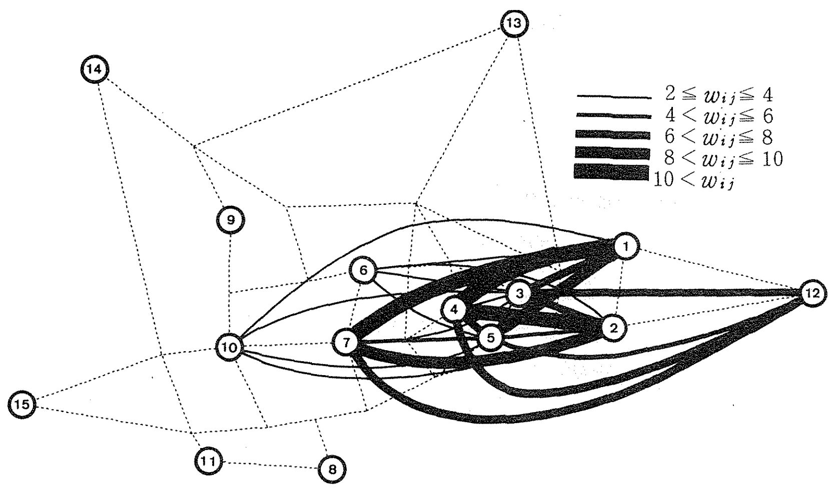
最初に配分対象リンク選択問題を解く。そのときに得られるトリップに対する重みは、図－6.32～6.34に示すようになった。これらの図から、着目リンク周辺に起終点を持つトリップに対しては、大きな重みが与えられていることがわかる。

配分対象リンク選択問題の解は、図－6.35～6.37に示すものである。ケース1とケース2のネットワークは同じになった。図－6.9の道路網に最短経路配分した結果が図－6.10であるが、配分対象リンク選択問題の解においては図－6.10で交通量の少ないリンクが除かれているのがわかる。そして、着目リンクに隣接するリンクが残される傾向にある。

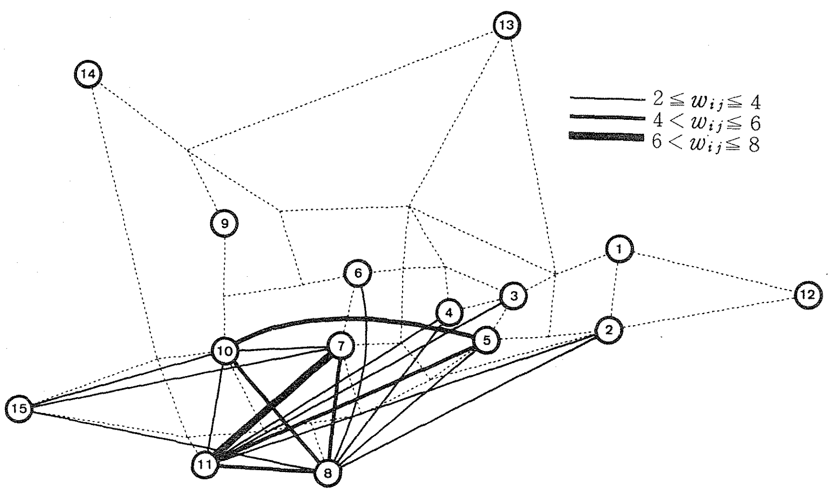
つぎに、リンクとノードの集約を行う。ケース1、2の解のネットワークでは、リンク23が孤



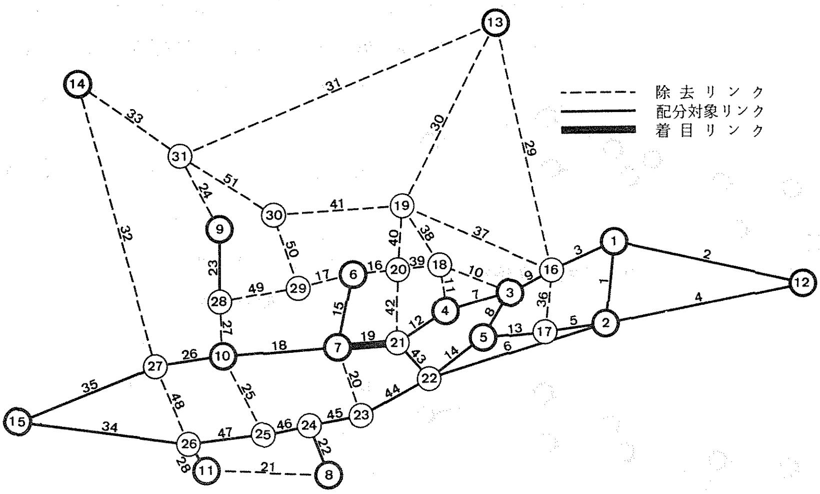
図— 6.32　ケース 1 のトリップに対する重み



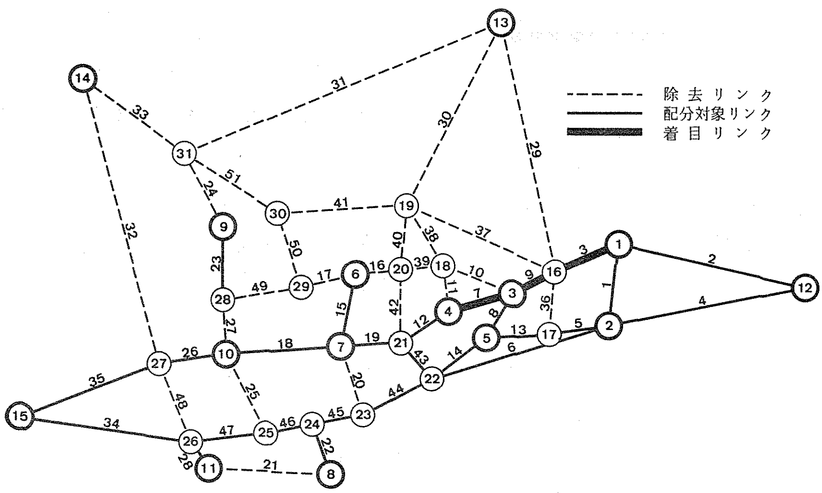
図— 6.33　ケース 2 のトリップに対する重み



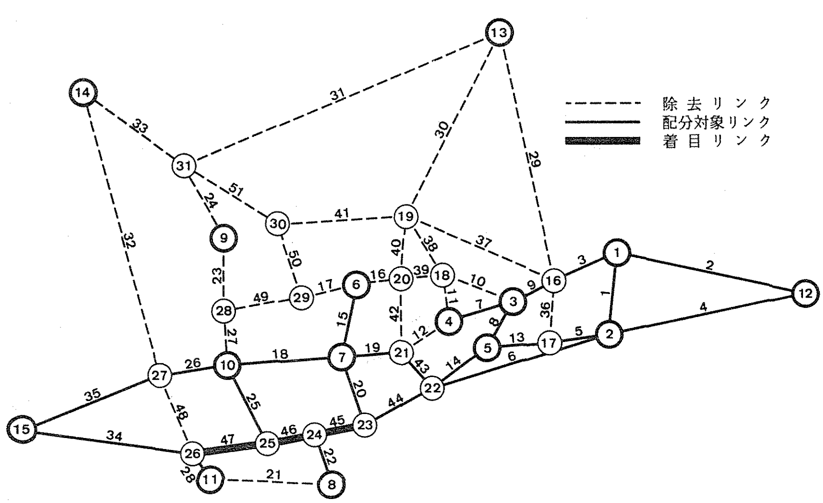
図— 6.34　ケース 3 のトリップに対する重み



図— 6.35　ケース 1 の配分対象リンク選択問題の解

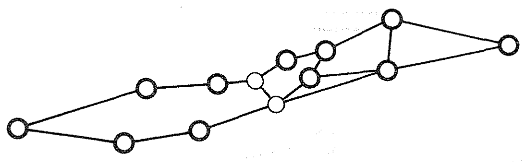


図— 6.36　ケース 2 の配分対象リンク選択問題の解

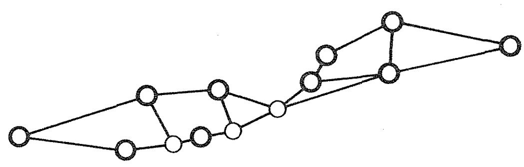


図— 6.37　ケース 3 の配分対象リンク選択問題の解

立している。そこで、リンク23を配分対象からはずす。そして、6.5.6で述べた方法を用いてネットワークを整えると、それぞれ図－6.38、6.39の配分対象道路網が得られる。詳細な道路網での発生・集中ノード数は15であったが、ケース1、2とケース3の配分対象道路網では11ノードと10ノードに減っている。全ノード数は31から13に、リンク数は51から17と18へ減っている。



図— 6.38　ケース 1, 2 の配分対象道路網



図— 6.39　ケース 3 の配分対象道路網

配分対象道路網が求まったので、そこに含まれるリンクの交通容量を決める。この場合、着目リンクの所要時間が実測されていれば、所要時間を固定して配分すればよい。しかし、ここでは着目リンクの交通量の推定値を比較するので、他のリンクと同じ方法で交通容量を決定する。

交通容量は、詳細な道路網と配分対象道路網での最短経路配分結果を用いて決める。それぞれの交通量で所要時間が等しくなるように交通容量を決めるのである。ここで使用するリンクの交通量と所要時間の関係式は、式（6.24）、（6.25）である。そのため、ふたつの配分結果でリンクの所要時間を等しくするには、リンク交通量の交通容量に対する割合を等しくすればよい。連続するリンクを集約して1本のリンクで表わしている場合、リンクの所要時間は元のリンクに分けて計算しなければならない。そして、リンクの往復両方向の交通容量は別に求める必要があるが、需要交通量がノード間の両方向で大差ないので往復両方向とも同じ交通容量とする。

交通容量を定めた後、配分対象道路網に等時間配分を行う。これにはIA*法[20]を用いる。すなわち、Phase 1で分割配分を行い、Phase 2で需要交通量の5％の再配分を20回繰り返すことにする。この方法では正確な等時間配分にはならないが、実際に用いられる簡単な分割配分法よりは配分の精度は高い。

配分結果からの詳細な道路網での交通量の推定は、リンク交通量の交通容量に対する割合が等しいと仮定して行う。推定結果は表－6.14になった。推定交通量は実測交通量に対する比率で表わしている。推定誤差の平均値は着目リンクが5.2％、着目リンク以外が16.1％である。全般にケース1、2の配分対象道路網の方が推定精度がよい。交通配分での精度からすれば、着目リンクだけでなく、それ以外のリンクの交通量の推定値も使用できる。

以上の計算例においては、期待する配分対象道路網が得られた。すなわち、ここでは詳細な道路網での配分結果が、簡単な配分対象道路網を使って推定できたのである。トリップに対する重みや配分対象リンク選択問題の定式化、それにリンクの交通容量の設定がすべて妥当であったといえる。

表一 6.14　リンク交通量の推定値

リンク	実測交通量	推定交通量の比率(%) ケース1，2	ケース3
1	1208	133.8	136.4
2	3609	105.7	71.9
3	3878	108.9*	80.7
4	6671	102.6	115.2
5	4759	96.5	107.4
6	3390	98.8	108.4
7	3241	100.6*	70.9
8	1635	109.3	122.6
9	3323	101.0*	74.8
12	2457	101.2	
13	4216	108.9	121.2
14	3993	112.6	119.8
15	885	134.8	134.8
18	4380	104.8	122.4
19	5196	95.4*	107.2
20	1063		92.1
22	4379	101.6	101.6
25	1968		28.7
26	4727	130.9	127.9
28	3560	98.1	98.1
34	3472	99.6	112.6
35	3122	99.3	97.0
43	2933	115.4	143.5
44	4603	122.2	120.9
45	5666	112.1	109.8*
46	3795	106.8	107.8*
47	4224	88.3	103.8*

*：着目リンク

6.5.9　むすび

ここでは、配分対象道路網を作成する手法を提案した。配分の対象とするリンクを選び出し、リンクとノードの集約を行い、最後に残されたリンクの交通容量を決めるものである。そして、配分対象リンク選択問題は最適交通網構成問題として定式化し、大規模問題を扱える近似解法を適用した。

計算例においては、詳細な道路網での着目リンクの交通量を、求めた配分対象道路網での配分結果から平均5.2%の誤差で推定できた。この結果は、配分対象リンク選択問題の定式化と解法、それにリンクの交通容量の決定方法が妥当であったことを示している。従来は試行錯誤で作成していた配分対象道路網を、容易に作成することができたのである。

簡単な例題の計算だけであるが、ここで提案した配分対象道路網の作成手法の有用性は示された。作成される道路網を用いた場合の配分交通量の精度については、なお検討を重ねることが必要である。ゾーン分割に対してバランスのとれた配分対象道路網を作成する問題に関しても、こ

の手法でどれだけ解決されているかを調べなければならないであろう。

6.6　バス路線網計画問題への適用[21)]

6.6.1　はじめに

バスの輸送機関としての長所のひとつは、路線や運行回数を変更し、容易に需要の変化に対応できることである。しかし、従来の路線設定や運行回数の変更作業は、もっぱら計画者の経験や勘によっている。そのため、改良は部分的な手直しにとどまることが多く、長所を生かして路線網全体を最適に維持あるいは計画することはほとんど行われなかった。

現在、バス事業の多くは赤字経営であり、経営改善のためにも路線網を需要に即したものにし、無駄をなくすとともに利用者に便利なものとする必要に迫られている。計算機を利用し、路線網の決定を合理的に、かつ容易に行えるシステムの開発が切に望まれているのである。

バス路線網計画において決定しなければならないのはバス道路網、路線系統、運行回数、配車計画、乗務員の勤務表である。これらは相互に関連しあっており、しかもそれぞれ決定の自由度が高い。そのため、全体としてのシステムをとらえ、合理的な目標を設定して最適なバスシステムを設計することは容易でない。そこで、ここではバスに対する需要が与えられたときに、バス道路網、路線系統、運行回数を決定する問題を取り上げ、これらを段階的に解くことにする。

バスを運行する道路網を決める問題は、一般的な最適交通網構成問題として解くことができる。また、路線系統の最適な組み合わせを求める問題も、最適交通網構成問題として扱える。したがって、ここで作成するバス路線網構成システムの主要部分には、最適交通網構成手法を応用する。

6.6.2では、バス路線網計画問題について述べ、提案するバス路線網構成システムの概要を示す。

6.6.3では、バスを運行する道路網を決めるためのバス道路網限定モデルについて述べる。

6.6.4では、検討の対象とする路線を列挙する路線列挙モデルについて述べる。

6.6.5では、総乗車時間が短くなる検討路線の組み合わせを求める路線網決定モデルについて述べる。

6.6.6では、総待ち時間が短くなるように路線の運行回数を決める運行回数決定モデルについて述べる。

6.6.7では、ここで提案するシステムを用い、バス路線網の再編成のケーススタディを行う。

6.6.2　バス路線網構成システムの概要

このシステムで決定するのは、運行するバス路線とその運行回数とする。そして、バス台数や路線数の制約下で、利用者に対して最適な路線網と運行回数を求めたいとする。利便性を総所要時間で表わすと、バス路線網計画問題の全体は、次のように表現できる。

問題6.7

$$\min \quad Z=\sum_{i}\sum_{j}\{t_{ij}^{1}(\boldsymbol{x},\boldsymbol{y})+t_{ij}^{2}(\boldsymbol{x},\boldsymbol{y})+t_{ij}^{3}(\boldsymbol{x},\boldsymbol{y})\}\,q_{ij}(\boldsymbol{x},\boldsymbol{y}) \qquad (6.75)$$

$$\text{s.t.}\quad \sum_k x_k \leqq M \tag{6.76}$$

$$\frac{1}{y_k}\sum_i \sum_j r_{ijkl}(\boldsymbol{x},\boldsymbol{y})\,q_{ij}(\boldsymbol{x},\boldsymbol{y}) \leqq C \tag{6.77}$$

$$\sum_k a_k x_k y_k \leqq B\cdot T \tag{6.78}$$

$$0 \leqq r_{ijkl}(\boldsymbol{x},\boldsymbol{y}) \leqq 1 \tag{6.79}$$

$$x_k = 0 \ or \ 1 \tag{6.80}$$

$$y_k = 0,\ 1,\ 2,\ \cdots\cdots \tag{6.81}$$

ここに、　Z : 総所要時間

t_{ij}^1 : i から j へのトリップのバス待ち時間

t_{ij}^2 : i から j へのトリップの乗車時間で、バスを使わないときは徒歩所要時間とする。

t_{ij}^3 : i から j へのトリップの乗換時間

q_{ij} : i から j へのトリップ数

$\boldsymbol{x} = \{x_1, x_2, \cdots, x_k, \cdots\}$

: 路線 k が路線網に含まれるとき $x_k = 1$、含まれないとき $x_k = 0$ とする。

$\boldsymbol{y} = \{y_1, y_2, \cdots, y_k, \cdots\}$

: 路線 k の運行回数を y_k とする。

r_{ijkl} : i から j へのトリップのうち路線 k の区間 l を通過するものの割合。

a_k : 路線 k の運行所要時間

M : 路線数の制約値

C : バスの定員

B : 運行可能バス台数

T : バス運行時間

式（6.75）の目的関数は総所要時間の最小化である。制約式は、式（6.76）が路線数の制約、式（6.77）が乗客数が定員以下という制約、式（6.78）がバス台数の制約である。この定式化においては、需要交通量や乗車経路は路線網と運行回数によって変化するとしている。

問題6.7の最適解を得るには、基本的にはすべての路線のあらゆる組み合わせに対して最適な運行回数を決定し、目的関数値を比較しなければならない。しかし、実際のネットワークでの実行可能な路線の数は膨大で、その組み合わせは無限といってもよい。この点が限られた数のリンクの組み合わせを調べる一般的な最適交通網構成問題と異なる点である。さらに、運行回数を決める問題だけを取り上げても非線形整数計画問題であり、解を得るにはかなりの計算を要する。これらのことから、たとえ若干の仮定を設けて問題を単純化しても、問題6.7をこのまま解くことは困難である。そこで問題を分割し、段階的に解くことにする。すなわち、まず可能な路線の中からいくらかの検討路線を選び、それらの中から最適な路線の組み合わせを作る。最後に路線網を構成する路線に対して最適な運行回数を決めるものとする。

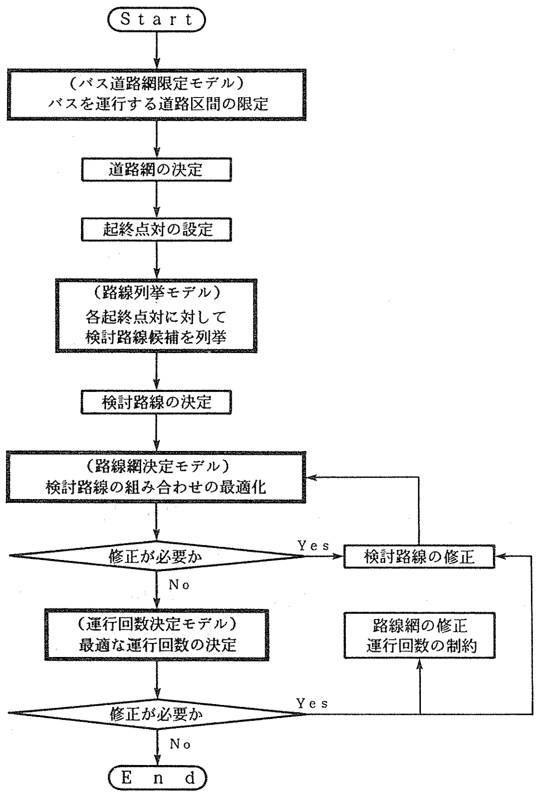
最適化問題を分割するため、および最適交通網構成手法を適用するために、次の仮定を設ける。

① 需要交通量は路線網、運行回数によって変化しない。

② 乗車経路は路線網によって決まり、運行回数によっては変わらない。

このシステムでは、バスの運行が可能なすべての道路区間でネットワークを構成し、そこに路線網を設定する。これらの仮定のもとで上に述べた段階的な方法で路線網を作ると、運行回数の少ない路線がネットワーク全体に広がることが起こり得る。これではフリークエントサービスの点からは魅力のない路線網になり、需要が計算に用いたものと大きく違ってしまう恐れがある。そこで、あらかじめバスを運行する道路区間を限定しておくことにする。

以上のようにすれば、バス路線網構成システムは、バス道路網限定モデル、路線列挙モデル、路線網決定モデル、運行回数決定モデルの4つのサブモデルから作られる。そして、各モデルの間には、人間が判断、修正を行い、フィードバックさせる部分をはさむ。これは、各モデルが路線網の評価要因をすべては考慮できないことと、ひとつの問題を分割した欠点を補うためである。また、主観的な判断を加えざるを得ない場合にも役立つ。システムのフローチャートは図－6.40



図－6.40 バス路線網構成システムのフローチャート

のようになる。

各モデルのうち、バス道路網限定モデルと路線網決定モデルは最適交通網構成手法を応用するものである。また、バス路線を列挙する問題は最適経路探索問題であり、最適交通網構成問題の一種と見なせる。バス路線網計画を一貫して行おうとする研究はあまりないが、最適交通網構成

手法を基礎にしていることがこのバス路線網構成システムの特徴である。

6.6.3 バス道路網限定モデル

バス道路網限定モデルは、対象地域のバスの運行が可能な区間のうち、路線の設置を検討する道路区間を決めるためのものである。このモデルの段階では、どのような路線にするかは決まっていない。しかし、路線を設けたときに需要に応じたものになる道路網でなければならない。そこで、総所要時間が最小になるネットワークを求める。ただし、路線については決まっていないので、待ち時間や乗換時間は考えない。

この問題が最適交通網構成問題であることは明らかである。停留所をノード、バスの運行が可能な道路区間をリンクとして最大ネットワークを作ればよい。そしてすべての停留所をバスで結ばなくてもトリップが目的地へ到達できるように、隣接停留所間の徒歩リンクを加えておく。バス道路網の長さを制約条件とすると、問題は次のようになる。なお、リンクの長さは所要時間で表わすものとする。

問題6.8

$$\min \quad Z=\sum_i \sum_j q_{ij} t_{ij}(\boldsymbol{x}) \tag{6.82}$$

$$\text{s.t.} \quad \sum_k l_k x_k \leqq L \tag{6.83}$$

$$x_k = 0 \ or \ 1 \tag{6.84}$$

ここに、Z：トリップの総所要時間

q_{ij}：ノード ij 間のトリップ数

t_{ij}：ノード ij 間の最短所要時間で、ネットワークにより異なる。

$\boldsymbol{x}=\{x_1, x_2, \cdots, x_k, \cdots\}$

：バスリンク集合 k がネットワークに含まれるとき $x_k=1$、含まれないとき $x_k=0$ とする。

l_k：バスリンク集合 k に属するリンクの長さの合計

L：ネットワークに含まれるバスリンクの長さの合計の制約値

問題6.8では個々のリンクでなく、リンクの集合の組み合わせを調べるように定式化している。これは、リンクの表わす道路区間ごとの取りはずしを検討する必要はなく、交差点間のようにいくつかのリンクをまとめた単位での取りはずしを検討すればよいからである。

問題6.8の特徴は、リンク集合で考えていることと、バスリンク以外に徒歩リンクを設けていることである。しかし、解法の上からは最適交通網構成問題の基本問題と大差ない。ここでは、局所最適化を行う近似解法を用いることにする。

このモデルで得られる解では、ターミナル以外の地点で道路がとぎれることがある。この場合、実際にはバスが方向転換できず、運行不可能である。このような点を修正するのが次の道路網決定のステップである。なお、こうして決めた道路区間にはバスが運行される可能性があるということを表わすだけである。選ばれる路線の組み合わせによっては、バスが運行されない道路区間が生じることもある。

6.6.4　路線列挙モデル

路線列挙モデルは、与えられた起点、終点の停留所間を結ぶ路線のうち、定められた路線評価関数の値の大きなものを列挙する機能をもつものである。ここでは、路線を列挙する方法と、路線の評価方法について述べる。

路線の起点と終点ではバスが方向転換できなければならない。そのため起終点になり得る停留所は限られる。そして、これらの方向転換が可能な任意の停留所間には路線を設定できるのであるが、明らかに路線を設ける必要のないものもある。そこで、不要な路線を探索する計算を省く。まずトリップ分布などを考慮して検討の必要のある起終点の組み合わせを作り、それぞれの起終点対にこのモデルを適用し、検討路線の候補を列挙するものとする。

路線列挙モデルでは、バス道路網限定モデルの結果を修正したネットワーク上で、与えられた起点から終点に至る路線を探索する。単に起終点間を結べばよいのであれば、路線は無数に存在する。しかし、非現実的な路線にバスを運行させることはできない。そこで、次に示す検討路線として必要な条件を設ける。これらの条件を満たす路線だけが実行可能とする。そして、実行可能な路線に対しては路線評価値を計算し、その値が大きければ検討路線候補とする。

① 路線長が指定される最短路線長以上、最長路線長以下である。

② 路線がループを含むとき、その長さが最長ループ長以下である。また同じノードを3回以上通らない。

③ あるノードから直前に通過したノードにもどらない。

④ 同じリンクを2回以上通らない。

①は、路線が短ければ両端での折り返しのための損失時間の割合が大きくてバスの運行効率が悪く、長すぎれば運行時刻が不正確になりやすいためである。②は、ループが路線に含まれるとループを1周させられる乗客ができるので、その長さを制限するものである。また同じ理由から同一の停留所を3回以上通ってはならないとし、複雑なループができないようにする。③は、任意の停留所で方向転換できないようにするためである。必要があれば、起終点となるノードではこの制約を設けない。④は利用者にわかりにくい複雑な路線を除くためである。ただし、このモデルではリンクはすべて有向リンクとし、同じ区間を逆に通るのはかまわないとする。

路線列挙問題はリンクの順列を作るもので、最適経路探索問題と考えることができる。解法には探索木を使う陰的列挙法が適用できる。この問題の場合、探索木の節点はネットワークのノードを表わし、枝は路線に採択されたリンクを表わすようにする。そして、探索木の出発点の節点は、路線の起点のノードに対応させる。ここから路線を延ばしてゆき、あらゆる実行可能な路線を調べるのである。

検討路線の制約条件に従った分枝規則を設けると、路線列挙モデルのアルゴリズムは次のようになる。

ステップ1：各ノードから終点ノードまでの最短所要時間 d_{it} を求め、$L_i^a \leftarrow L^{max} - d_{it}$ とする。

ステップ2：初期値として $L \leftarrow 0,\ i \leftarrow s,\ r \leftarrow 2,\ x_k \leftarrow 0(k=1,\ 2,\ \cdots,\ m),\ R_1 \leftarrow 0,\ y_j \leftarrow 0(j=1,\ 2,\ \cdots,\ n,\ j \neq s),\ y_s \leftarrow 1$ とする。

ステップ3：$k \leftarrow A_i$ とする。

ステップ４：$x_k = 1$あるいは$b_k = R_{r-1}$ならステップ11へ進む。

ステップ５：$i \leftarrow a_k$とし、もし$L + l_k > L_i^a$ならばステップ11へ進む。

ステップ６：$y_i = 0$ならステップ８へ、$y_i = 2$ならステップ11へ進む。

ステップ７：ループ長を求め、最長ループ長を超えていればステップ11へ進む。

ステップ８：$L \leftarrow L + l_k$, $R_r \leftarrow k$, $r \leftarrow r + 1$, $x_k \leftarrow 1$, $y_i \leftarrow y_i + 1$とする。もし$i \neq t$ならステップ３へもどる。

ステップ９：$L < L^{min}$ならステップ３へもどる。

ステップ10：$R_2, \cdots, R_{r-1}$で表わされる路線に対して路線評価値を計算する。すでに求められている路線とこの路線の中で、評価値の大きいものN個を記憶する。ステップ３へもどる。

ステップ11：$k \leftarrow B_k$とし、$k > 0$ならステップ４へもどる。

ステップ12：$r \leftarrow r - 1$とする。$r = 1$ならステップ14へ進む。

ステップ13：$k \leftarrow R_r$, $x_k \leftarrow 0$, $L \leftarrow L - l_k$, $i \leftarrow a_k$, $y_i \leftarrow y_i - 1$とし、ステップ11へもどる。

ステップ14：記憶されている路線を評価値の順に出力し、計算を終える。

ここに、d_{ij}：ノードij間の最短所要時間

L_i^a：ノードiにおける許容路線長

L^{max}：最長路線長

L^{min}：最短路線長

L：路線長

R_r：路線の通過リンク番号

A_i：ノードiを起点とするリンクのうち、最小のリンク番号

B_k：リンクkと同じノードを起点とするリンクのうち、kの次に小さいリンク番号。該当するリンクがなければ$B_k = 0$とする。

a_k：リンクkの終点のノード

b_k：リンクkと起点、終点が逆のリンク

l_k：リンクkのリンク長

x_k：リンクkが路線に含まれているとき１、含まれていないとき０とする。

y_i：ノードiの通過回数

N：列挙路線数

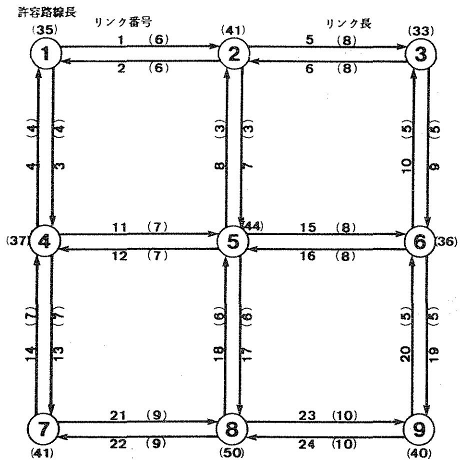
s：路線の起点ノード

t：路線の終点ノード

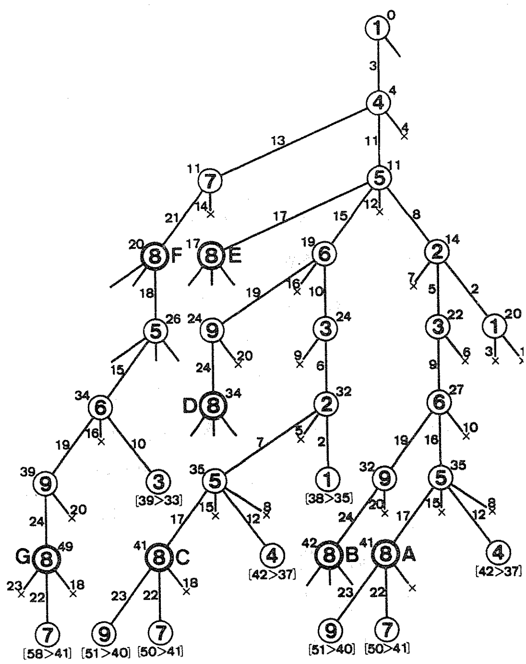
m：リンク数

n：ノード数

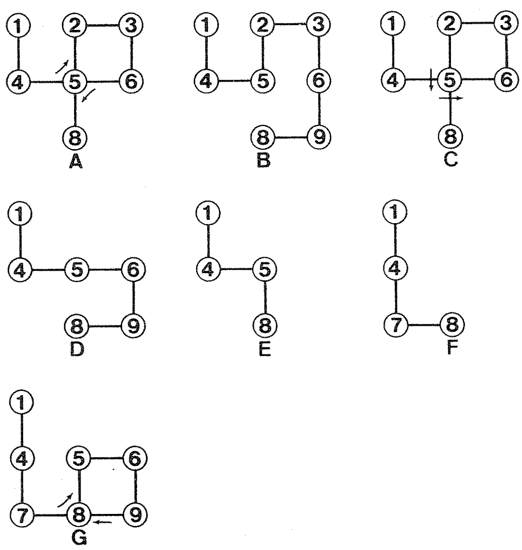
このアルゴリズムによる計算例として、図－６.41のバス道路網を用いて路線を列挙する。起点をノード１、終点をノード８、最短路線長を10、最長路線長を50、最長ループ長を30とする。リンク長と各ノードでの許容路線長は図－６.41に示すとおりである。そして、アルゴリズムに従って計算すると、探索木は図－６.42のようになる。この図には、リンク番号、ノード番号、路線長も示す。探索木の図の部分で列挙された路線は図－６.43に示すものである。



図― 6.41　バス道路網

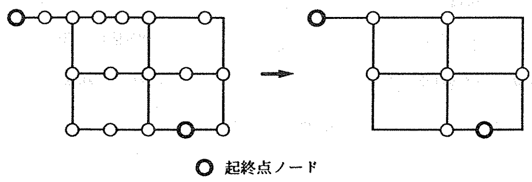


図― 6.42　路線の探索木



図― 6.43　列挙された路線

路線の列挙方法は以上のとおりであるが、計算例からもわかるように、分岐の起こらないノードは路線を探索するネットワークから除いてよい。そこで、集約したネットワークで路線を探索し、計算時間を短縮する。ネットワークの集約は、図－6.44のように不要なノードを除き、リンクをまとめるものである。



図－6.44　ネットワークの集約

列挙した路線の良否を判断する指標として路線評価関数を設ける。路線を評価する基準は多くあるが、次の基本的な条件を満たしていることが運行者と利用者の双方に望ましいであろう。

① 利用する乗客が多いこと

② 各停留所間の最短経路を成していること

乗客が多ければ収益は多く、利用者の需要に応えていることになる。そして、最短経路を成していれば乗客は早く目的地に着け、運行者にはバスの運行効率がよい。そこで、これらの条件を盛り込んだ路線評価関数を作成し、検討路線候補を列挙する。そのうえで、区間別、方向別の乗客数の差などの他の評価基準を考慮して検討路線を決定する。

乗客数の表わし方は何通りかあるが、どれだけの需要を満たしているかという点からは、総乗車時間あるいは総乗車距離が適している。しかし、遠回りして乗車している場合、総乗車時間には迂回した乗車時間が含まれる。満たした需要を調べるには、この分を差し引かなくてはならない。そこで、その路線での乗車時間ではなく、最低限乗らなければならない時間を使う。すなわち、乗客の起終点間の最短所要時間を合計する。そして、これだけでは路線を長くするほどよいことになるので、単位路線長当りの値を使用する。式に表わせば、次のようになる。

$$F_1 = \frac{\sum_i \sum_j q_{ij}\, d_{ij}^0}{L} \qquad (6.85)$$

ここに、 q_{ij} ：ノードij間のトリップ数

d_{ij}^0 ：ノードij間の最短所要時間

L ：路線長

i, j ：路線上のノード

②の条件については、総乗車時間に対する最短所要時間の合計の割合で表わし、式(6.86)とする。

$$F_2 = \frac{\sum_i \sum_j q_{ij}\, d_{ij}^0}{\sum_i \sum_j q_{ij}\, d_{ij}} \qquad (6.86)$$

ここに、 d_{ij} ：ノードij間の乗車時間

式(6.85)、(6.86)より、路線評価関数を式(6.87)のように定める。この評価関数の値が大きいほどよい路線であるとする。

$$F = F_1 \cdot F_2 = \frac{(\sum_i \sum_j q_{ij}\, d_{ij}^0)^2}{L \sum_i \sum_j q_{ij}\, d_{ij}} \qquad (6.87)$$

6.6.5 路線網決定モデル

路線列挙モデルで検討路線候補を列挙し、それらから検討路線を決定した後、このモデルで路線の組み合わせを決める。ここでは問題を定式化し、その近似解法を示す。

システム全体の目的関数には、総所要時間の最小化を考えていた。需要交通量と乗車経路に関する仮定を設けたので、所要時間のうちの乗車時間は路線の組み合わせだけで決まる。これに対し、待ち時間と乗換時間はバス運行回数が決まらなければ求まらない。しかし、乗車経路は運行回数によって変化しないとしているので、乗換回数は路線網で決まる。そこで、どの検討路線を路線網に含めるかを決めるにあたっては、総乗車時間が短く、かつ乗換回数を少なくする。定式化すれば、路線網決定問題は次のようになる。

問題6.9

$$\min \quad Z=\sum_i\sum_j q_{ij}\{(d_{ij}+rt_{ij})y_{ij}+(d_{ij}^0+p)(1-y_{ij})\} \qquad (6.88)$$

$$\text{s.t.} \quad \sum_k x_k \leqq M \qquad (6.89)$$

$$\sum_i\sum_j q_{ij}(d_{ij}^0+p)(1-y_{ij}) \leqq P \qquad (6.90)$$

$$t_{ij} \leqq N \qquad (6.91)$$

$$x_k = 0 \ or \ 1 \qquad (6.92)$$

$$y_{ij} = 0 \ or \ 1 \qquad (6.93)$$

ここに、 Z : 総乗車時間と乗換抵抗、経路がないときのペナルティーの和

q_{ij} : ノード ij 間のトリップ数

d_{ij} : ノード ij 間の最短所要時間

d_{ij}^0: 最大ネットワークにおけるノード ij 間の最短所要時間。ただし、最大ネットワークで経路がないときは $d_{ij}^0=0$ とする。

t_{ij} : ノード ij 間の乗換回数

y_{ij} : ノード ij 間に経路があるとき1、経路がないとき0とする。

x_k : 路線 k がネットワークに含まれるとき1、含まれないとき0とする。

r : 乗換抵抗

p : 経路がないときのペナルティー

M : 路線数の上限値

P : ペナルティーの上限値

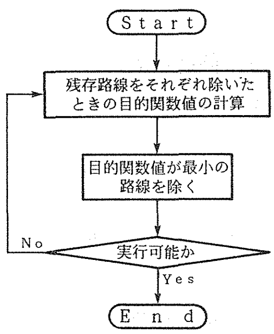
N : 乗換回数の制約値

式(6.88)は、総乗車時間と乗換抵抗、経路がないときのペナルティーの和の最小化を表わす目的関数である。この式の d_{ij} 、t_{ij} 、y_{ij} はいずれも路線網によって決まる。経路がないときのペナルティーを設けたのは、場合によってはバスを利用できないトリップが生じてもよいとするためである。式(6.89)はネットワークに含まれる路線数に対する制約である。式(6.90)は、経路がないときのペナルティーに上限値を設け、バスを利用できないトリップの増加を制限するものである。上限値の与え方によっては、全トリップがバスを利用可能にすることもできる。式(6.91)は乗換回数を N 回以内とするもので、それ以上必要ならばそれらのノード間に

経路はないものとする。また乗車経路は、乗換回数が最も少ない経路の中から最短所要時間の経路を選ぶものとする。

路線網決定モデルの適用にあたり、他の検討路線に完全に含まれる検討路線は目的関数値の改善に寄与しないことに注意しなければならない。問題6．9では路線長やバスの運行に関する制約を設けていない。そのため、他の検討路線に含まれる路線を採用するかどうかは、ここで決められない。そこで、他の検討路線に完全に含まれることのない検討路線だけをこのモデルの対象とする。そして、得られた路線網を構成する路線とそれらに含まれる路線に対して運行回数を決める。

問題6．9は最適な路線の組み合わせを求める問題である。この点でリンクの組み合わせを求める通常の最適交通網構成問題と異なる。しかし、解法の基本的な考え方は変わらない。ここで解くのはひとつの問題を分割、単純化した一部であり、対象とするネットワークの規模は大きい。したがって、簡単な近似解法が適している。backward 法を使うことにすると、計算手順は図－6．45のようになる。すべての路線を含んだ最大ネットワークから出発し、除いたときの目的関数値の増加が最小となる路線を除く操作を実行可能になるまで繰り返す。



図－6.45　路線網決定モデルの計算手順

目的関数の計算では、乗車時間を求めるために乗り換えを考慮した最短経路問題を解かなければならない。このとき、停留所をノードとしているためにトリップのない起終点対が多いこと、乗り換えを行うノードは一部だけであることに注意する必要がある。乗車時間と乗換回数は、次のアルゴリズムで求める。

ステップ１：初期値として、$n \leftarrow 0$、すべてのijについて$d_{ij} \leftarrow \infty$, $y_{ij} \leftarrow 0$とする。

ステップ２：$q_{ij} > 0$のijに対し、

$$d_{ij} \leftarrow \min_k \{ d_{ijk} \mid x_k = 1,\ d_{ijk} < \infty \} \qquad (6.94)$$

によって d_{ij} を決める。もし d_{ij} が決まれば $t_{ij} \leftarrow 0$, $y_{ij} \leftarrow 1$ とする。すべての$q_{ij} > 0$のijについて$y_{ij} = 1$になればステップ８へ進む。

ステップ３：$i \in T$, $q_{ij} = 0$のijに対し、式（6.94）によってd_{ij}を決める。d_{ij}が決まれば $t_{ij} \leftarrow 0$,　$y_{ij} \leftarrow 1$ とする。

ステップ４：$n \leftarrow n+1$とする。

ステップ５：$q_{ij} > 0$, $y_{ij} = 0$のijに対し、

$$d_{ij} \leftarrow \min_h \{ d_{ih} + d_{hj} \mid h \in T,\ t_{ih} = 0,\ t_{hj} = n-1 \} \qquad (6.95)$$

によってd_{ij}を決める。もしd_{ij}が決まれば $t_{ij} \leftarrow n$, $y_{ij} \leftarrow 1$ とする。すべての$q_{ij} > 0$のijについて$y_{ij} = 1$になればステップ８へ進む。

ステップ６：もし$n = N$ならばステップ８へ進む。

ステップ７：$i \in T$, $q_{ij} = 0$, $y_{ij} = 0$のijに対し、式（6.95）によってd_{ij}を決める。d_{ij}が決まれば $t_{ij} \leftarrow n$, $y_{ij} \leftarrow 1$とする。ステップ４へもどる。

ステップ8：式（6.88）によって目的関数値を求める。ただし、式（6.90）が満たされていないときは$Z \leftarrow \infty$とする。

ここに、d_{ijk}：路線kによるノードij間の所要時間

T：乗り換えが可能なノードの集合

n：乗換回数

ステップ2は直通路線の探索で、ステップ5は乗換回数がn回の経路の探索である。ステップ3とステップ7は、乗り換えが必要な経路を探す準備のために、乗換可能ノードから他のすべてのノードまでの経路を探すものである。経路がないときのペナルティーの値は、目的関数と式（6.90）で同じである。そこで、ステップ8で解が式（6.90）を満たすかどうかの検査も行う。その結果、求められた目的関数値が無限大でなければ、その解は路線数以外の制約条件を満たしていることになる。

目的関数の計算は以上のように行うが、backward 法で除くのは1路線ずつなので、すべての経路を求め直す必要はない。除く路線に関係する部分だけを計算し直せばよい。

6.6.6 運行回数決定モデル

路線網決定モデルで求めた路線網を構成する路線とそれらに含まれる路線に対し、このモデルで運行回数を決める。問題の定式化に先立ち、乗客の行動について次の仮定を設ける。

① 乗車経路は運行回数にかかわらず一定とする。

② 乗車経路は乗換回数が最小の経路の中で所要時間が最も短いものとする。

③ 乗り換えられる停留所がいくつかあるときは、最初の路線にできるだけ長く乗るものとする。

④ 複数の並行する路線が利用可能なとき、各路線に乗車する割合は運行回数に比例するものとする。

⑤ 停留所での待ち時間は、利用できるすべての路線の平均運行間隔の½とする。

⑥ 乗換時間は乗り換える停留所での待ち時間とする。

①、②はこれまでのモデルでも使ってきたものである。③、④は区間別の乗客数を明確にするためのものである。⑤、⑥は待ち時間と乗換時間を定義するものである。

乗車経路を一定とすると乗車時間も変化しない。そこで、ここでは待ち時間と乗換時間を最小化する。仮定により、乗車経路と乗換停留所はあらかじめ決まり、乗換時間を待ち時間と同じように扱える。そのため、乗り換えを行うトリップは分割して考えることができる。たとえば、ノードhで乗り換えるノードij間のトリップは、ノードih間とノードhj間のトリップに分ければよい。このような処理を行えば、トリップはすべて乗り換えを必要としないものだけにできる。そして、問題の定式化においては乗り換えを考慮しなくてよい。目的関数は総待ち時間の最小化だけとなる。

制約条件は、各路線の区間ごとの乗客数が輸送定員を上回らないことと、必要なバス台数が運行可能バス台数以下であることとする。運行回数決定問題は、次のようになる。

問題6.10

$$\min \quad Z = \frac{T}{2} \sum_{i} \sum_{j} \frac{q_{ij}}{\sum_{k} \delta_{ijk}\, y_k} \qquad (6.96)$$

s.t.
$$\sum_i \sum_j \frac{r_{ijkl}\, q_{ij}}{\sum_h \delta_{ijh}\, y_h} \leqq C \tag{6.97}$$
$$\sum_k a_k\, y_k \leqq B \cdot T \tag{6.98}$$
$$y_k = 0,\ 1,\ 2,\ \cdots \tag{6.99}$$

ここに、 Z：総待ち時間

q_{ij}：ノードij間のトリップ数で、乗り換えを処理したものとする。

δ_{ijk}：ノードij間のトリップが路線kを利用するとき1、利用しないとき0とする。

r_{ijkl}：ノードij間の乗車経路に路線kの区間 が含まれるとき1、含まれないとき0とする。

y_k：路線kの運行回数

a_k：路線kの往復運行所要時間

T：バス運行時間

C：バスの定員

B：運行可能バス台数

式（6.96）は総待ち時間の最小化を表わす目的関数である。式（6.97）は定員に関する制約条件を表わし、式の数は延べ区間数だけある。式（6.98）はバス台数の制約である。

問題6.10は定員に関する制約式が非常に多いため、このまま解くことは容易でない。しかし、式（6.97）の左辺は目的関数と似た形をしており、式（6.97）の左辺の値を小さくすれば目的関数値も小さくなる。すなわち、式（6.97）は式（6.98）のバス台数の制約に比べれば緩い制約である。そこで、定員に関する制約を省いて解を求め、必要ならば再計算を行う方法で実用的には十分である。そうすれば、問題はバス台数に関する制約式だけの簡単なものになる。

式（6.99）では運行回数は非負の整数値をとれるようにした。だが、実際には乗客に便利なように運行間隔が3分、5分、10分というように決められることが多い。これに対応して、ここでも運行間隔がきれのよい数字になるように決めることにする。そのため、運行間隔と運行回数をたとえば**表－6.15**のようにサービスレベルの関数として決めておく。そして、各路線のサービスレベルを決定する問題に変形する。

表―6.15 運行サービスレベルの例

サービスレベル	運行間隔（分）	運行回数（回／時間）
1	2	30
2	3	20
3	4	15
4	5	12
5	6	10
6	10	6
7	12	5
8	15	4
9	20	3

問題は非線形整数計画問題であるが、解法には最適交通網構成問題の forward 法に相当する近似解法を用いることにする。計算手順は次のとおりである。

ステップ1：すべての路線のサービスレベルを最低の状態にする。

ステップ2：各路線について、その路線のサービスレベルを1段階上げたときの所要バス台数と総待ち時間を調べる。ただし、サービスレベルが最高になっていれば、それ以上は上げない。

ステップ3：サービスレベルを1段階上げたときに所要バス台数が運行可能バス台数以下で、所要バ

ス台数の増加量に対する総待ち時間の減少量の割合が最大の路線のサービスレベルを1段階上げ、ステップ2へもどる。どの路線のサービスレベルを上げても運行可能バス台数以上になるときは計算を打ち切る。

6.6.7 バス路線網再編成のケーススタディ

ここでは、提案したバス路線網構成システムを実際のバス路線網再編成問題に適用し、その実用性を示すとともに路線網最適化の効果を調べる。最初に対象とするバス路線網再編成問題を示す。つぎに、バス路線網構成システムの計算手順に従い、バス道路網の決定、検討路線の決定、路線網の決定、運行回数の決定の順序で述べる。そして、最後に作成した路線網と現在の路線網とを比較する。

対象とする問題は尼崎市内のバス路線網を再編成するものである。より効率的な路線網にするために、次の各項目に従って路線網計画案を策定する。

① 不要な路線を整理し、路線数を減らす。

② 長い路線を減らす。

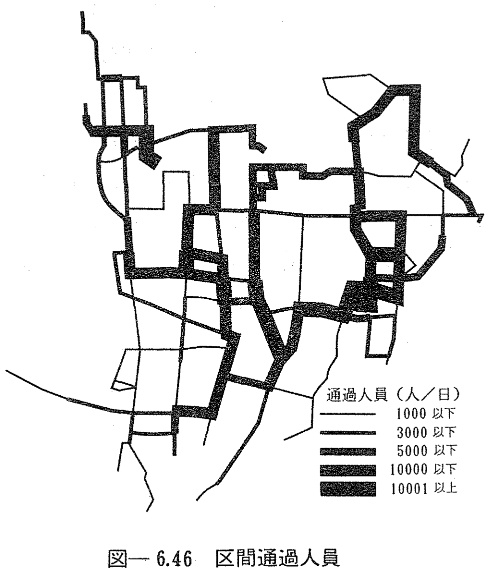
③ 必要なバス台数を減らす。

④ バスを運行する区間には十分な運行回数を確保する。

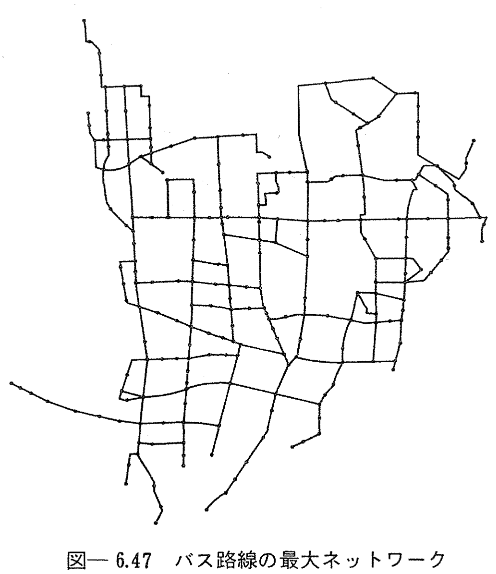
⑤ 乗客の所要時間ができるだけ長くならないようにする。

需要交通量は、昭和50年に乗客の流動調査で求められた停留所間OD表をそのまま使うことにする。したがって、この時点での状態を現在の路線網とし、作成した路線網案をこれと比較する。この路線網の営業キロは104km で、79系統に平日1971本のバスが運行されている。現在の路線網での乗客の流れは、図－6.46に示すとおりである。

少ないバス台数で運行回数を確保するため、バス道路網限定モデルを適用する。対象地域のバスを運行可能な道路は図－6.47に示すとおりで、ノードは既存の停留所あるいはそれ以外の交



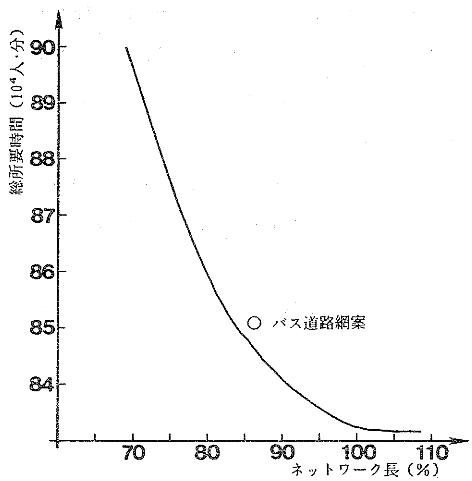
図－6.46 区間通過人員



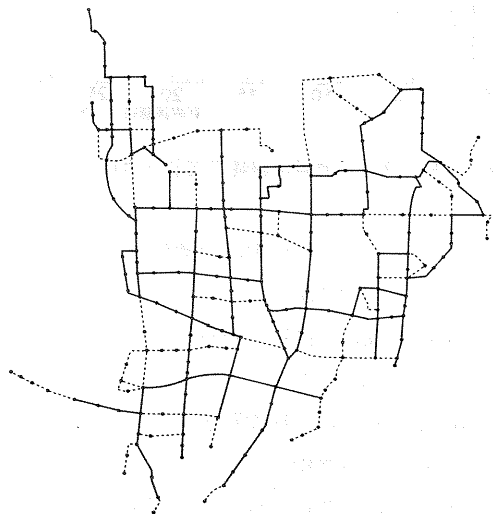
図－6.47 バス路線の最大ネットワーク

差点を表わす。これに隣接停留所を結んだ徒歩ネットワークをあわせ、最大ネットワークとする。

バス道路網限定モデルによる計算の結果、バス道路網のネットワーク長と総所要時間との関係は図－6.48のようになった。ただし、ネットワーク長は現在のネットワーク長に対する割合で示してある。徒歩の80m／分に対してバスの平均速度は290m／分であることから総所要時間の増加は比較的小さい。ネットワーク長を現在の70％にしても約8％増加するだけである。70％の場合のネットワークを例として図－6.49に示す。

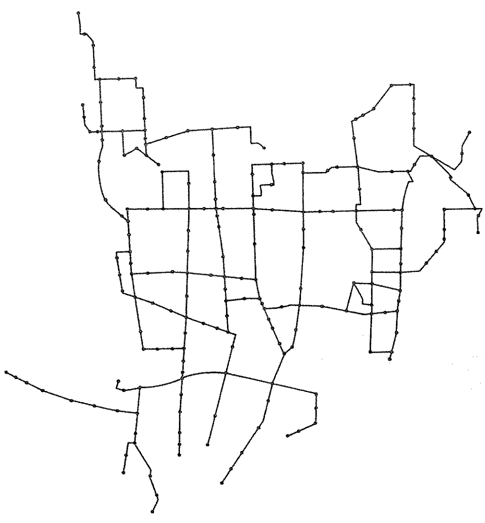


図－6.48　バス道路網限定モデルの計算結果



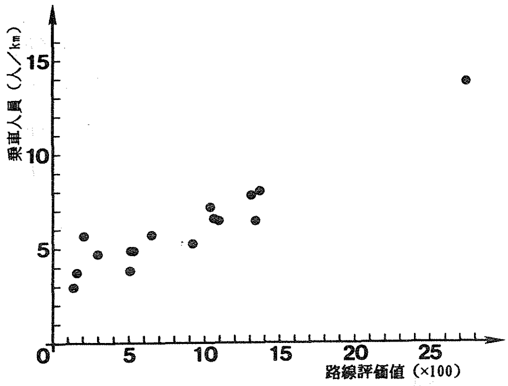
図－6.49　ネットワーク長を70％にしたときの解

後に続く路線列挙モデル以降の各段階でもネットワークが縮小される可能性があるので、ここではネットワーク長を現在の80％に制約したときの解を採用する。そして、方向転換できない停留所で行き止まりになっている個所を修正し、図－6.50のようにバス道路網を定める。このネットワークのバスリンクの長さの合計は現在の86.2％で、総所要時間は85×10^4人・分である。



図－6.50　バス道路網案

つぎに路線列挙モデルを適用するが、その前に路線評価関数の妥当性を確認しておく。図－6.47の道路網で現在の各路線の路線評価値を計算すると、乗車人員の実績値との関係は図－6.51に示すようになる。この図では、乗車人員の資料にあわせるために評価値を線別の値に集計している。評価値の計算においてはほかに路線はなく、路線上にトリップの起終点があれば必ずその路線を利用するとしている。しかし、図－6.51をみれば、多くの路線が存在する場合でも評価値の大きい路線の乗車人員は多い。



図― 6.51　現在の路線の路線評価値と乗車人員

式（6.87）の評価関数で検討路線を決めることに問題はないと考えられる。

図－6.50のネットワークにはバスが方向転換できるノードは27ある。これらの組み合わせのうち、起終点間の最短所要時間が25分以下で、現在の路線の起終点かあるいは需要交通量の分布から検討の必要があると認められるものを選び、路線を列挙する起終点対とする。最短所要時間を25分以下とするのは長い路線を作らないためで、距離に直せば7～8 km になる。

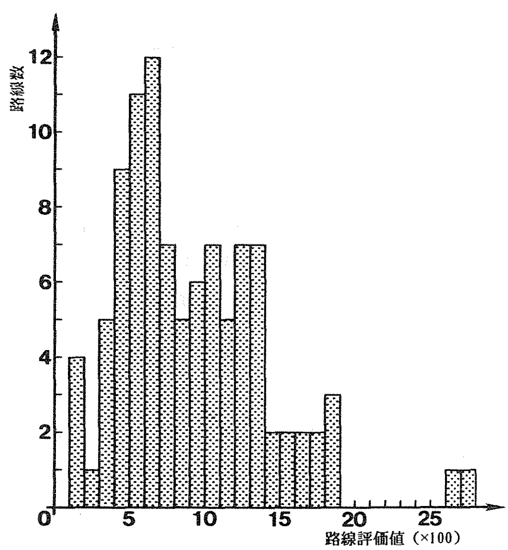
このようにして選んだのは97個の起終点対で、そのうちの39個が現在の路線に対応するものである。これらに路線列挙モデルを適用するときのパラメータは次のようにする。

① 最長路線長は最短所要時間の1.5倍とし、最短所要時間が10分未満のときは最短所要時間に5分を加えたものとする。ただし、循環路線では30分とする。

② 最長ループ長は循環路線を除き5分とする。

③ 極端に短い路線はないので最短路線長は制約しない。

これらの条件で路線を列挙し、評価値の大きさや他の路線との重複、区間別乗客数を考慮して61個の起終点対の99路線を検討路線とした。検討路線の路線評価値の分布を図－6.52に示す。

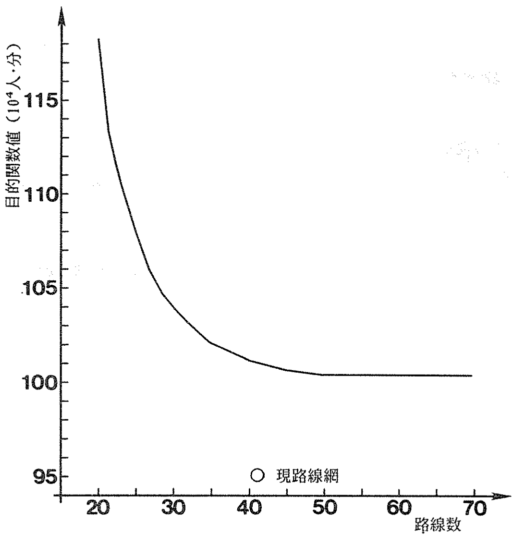


図― 6.52　検討路線の路線評価値の分布

99路線のうち他の検討路線に含まれるものを除き、70路線に対して路線網決定モデルを適用す

る。パラメータは乗換抵抗を10分、経路がないときのペナルティーを40分、その上限値は約1000トリップがバスを利用できない値である80,000人・分とする。ここでは時間帯によって路線網を変化させることはせず、全日の需要交通量を用いて路線網を決める。

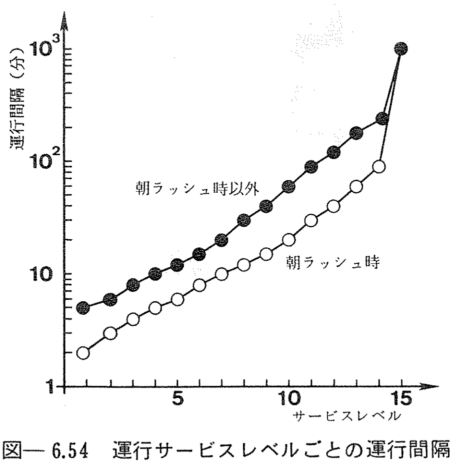
計算の結果は図－６.53のようになった。ネットワークが異なるので正確な比較はできないが、



図— 6.53　路線網決定モデルの計算結果

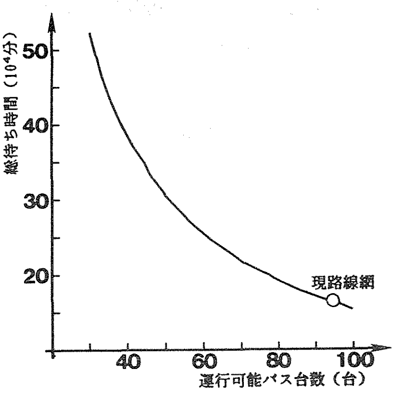
最大ネットワークでも現在の路線網の目的関数値を5.6%上回っている。これは、バスを走らせる道路区間を限定したためである。目的関数値があまり大きくならない範囲で路線を減らし、35路線の路線網を採択する。この場合、目的関数値は現在より7.4%増加する。

採択した路線網を構成する路線とそれらに含まれる検討路線の合計58路線の運行回数を決める。ここでは、午前７時から８時30分までの朝のラッシュ時と、それ以外の時間帯に分けて計算する。運行サービスレベルは図－６.54のようにする。

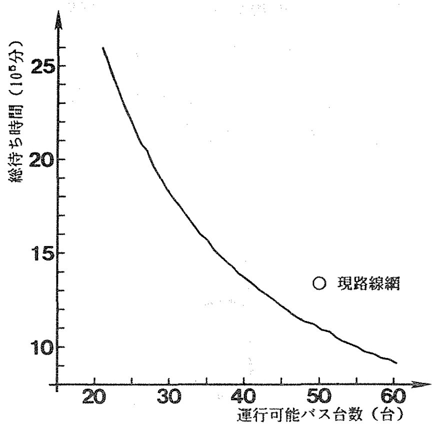


図— 6.54　運行サービスレベルごとの運行間隔

計算結果は、それぞれ図－６.55、６.56のようになった。バス台数を現在と同じにすれば、総待ち時間はラッシュ時で1.1%、それ以外で18.5%、全日をあわせれば16.6%短くなる。路線を減らし、長大路線をなくしたので乗換回数が増加しているにもかかわらず、現在よりも待ち時間は短くなっている。これは、運行回数の最適化の効果が大きいことを示している。現在の路線網に対して運行回数決定モデルを適用すれば、ラッシュ時で31.0%、それ以外では42.0%待ち時間が減少することが、これを裏付けている。ただし、現在の



図— 6.55　運行回数決定モデルの計算結果
（朝ラッシュ時）

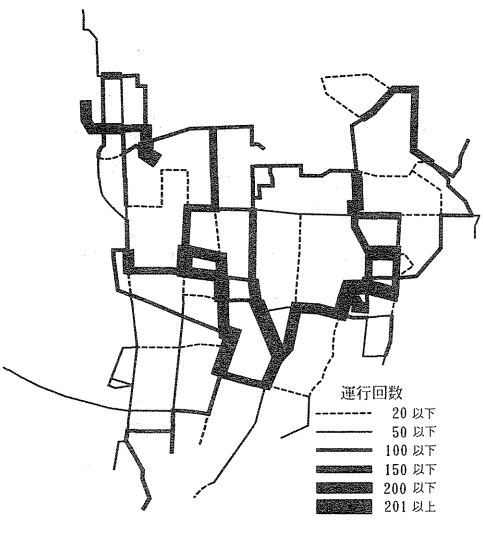


図— 6.56　運行回数決定モデルの計算結果
（朝ラッシュ時以外）

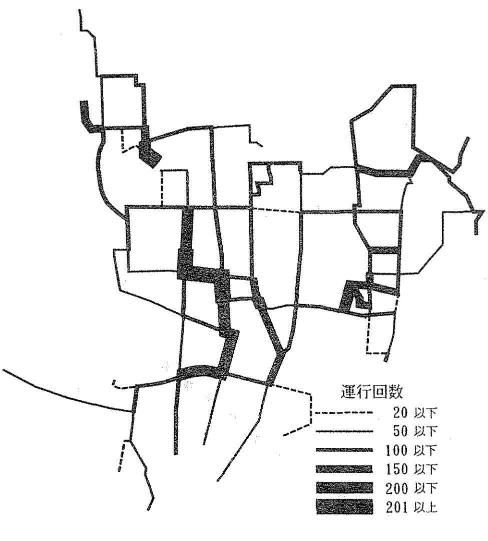
路線網の現行の運行回数では乗車経路選択の仮定が現実と合わず、待ち時間の計算値が大きくなっていると考えられる。その場合には待ち時間は計算値よりも短くても、乗車時間は長いことになる。したがって、運行回数決定モデルで適切な運行回数が決められていることはまちがいない。

計画案として、運行可能バス台数を現在の90%にしたときの解を採用する。総待ち時間はラッシュ時で10.2%現在より長くなるが、ラッシュ時以外で9.8%、全日では7.6%短くなる。また各路線の区間別乗客数を調べた結果、輸送定員を大きく超えている区間はない。

現在の路線網と計画した路線網での区間別運行回数は、図－６.57、６.58のとおりである。

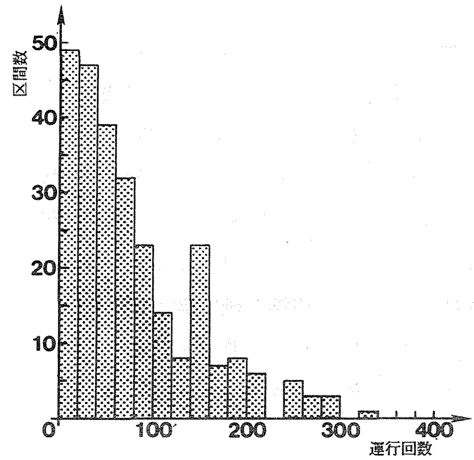


図— 6.57　現在の路線網の区間別運行回数

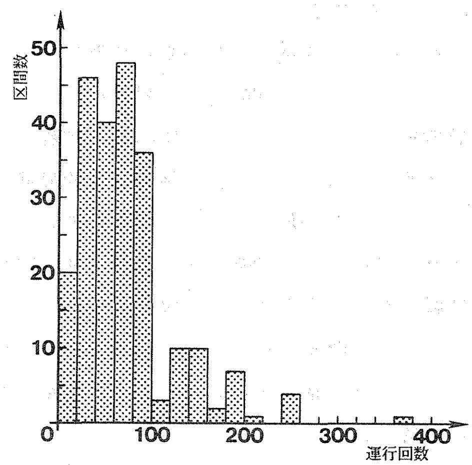


図— 6.58　計画路線網の区間別運行回数

そして、それぞれの路線網での区間別運行回数の分布を示したものが図－６.59、６.60である。これらを見れば、計画した路線網では運行回数が極端に少ない区間が減り、区間ごとのばらつき



図— 6.59　現在の路線網の区間別運行回数の分布



図— 6.60　計画路線網の区間別運行回数の分布

が小さくなっていることがわかる。

現路線網と計画路線網とをいくつかの項目で比較すると、表－6.16のようになる。バス台数

表— 6.16　計画路線網と現路線網との比較

項目	現路線網	計画路線網
ネットワーク長	353.5分	296.5分
停留所数	209	184
路線数	76	44
所要バス台数	ラッシュ時　93.3台 それ以外　49.5台	ラッシュ時　83.9台 それ以外　44.5台
総運行回数	1971回	1852回
平均区間別運行回数	81.2回	74.1回
総乗換回数	8回	10136回
徒歩を要するトリップ数	0	2944
平均所要時間	23.8分	23.7分
平均待ち時間	14.0分	14.3分

を減らしたために運行回数が減り、路線数を減らしたために乗換回数が増加している。そして、停留所がなくなり近くの停留所まで歩かなければならないトリップが全体の３％ある。しかし、待ち時間の増加はわずかで、徒歩時間と待ち時間、乗車時間を合計した平均所要時間は現在の路線網の値を0.1分下回っている。

このシステムで求めた計画路線網は、現路線網よりもネットワークは短く、路線数と所要バス台数も少ないが、利用者の所要時間はほぼ同じである。そして、運行回数の少ない区間は減っている。この結果から、最初に上げた項目を満たす路線網計画案を作ることができたといえる。

6.6.8　むすび

バス路線の設定や運行回数の変更作業は、これまで計画者の経験と勘によって行われてきた。ここでは、それらを合理的かつ容易に行うためのシステムを開発した。このシステムで決めるのは、路線網とそれを構成する路線の運行回数であるが、問題の複雑性からこれらを同時に決定するのは困難である。そこで、全体を部分的最適化問題に分割した。すなわち、バスに対する需要交通量とバス保有台数が与えられた条件のもとで、まずバスを運行する道路網を限定し、検討路線を列挙して最適な路線網を求めた後、運行回数を決定するものとした。これに従い、対象地域内で路線の設置を検討する道路区間を決めるバス道路網限定モデル、路線評価関数の値の大きい路線を列挙する路線列挙モデル、総所要時間が短く、かつ乗換回数が少ない路線を求める路線網決定モデル、待ち時間が短くなるように路線の運行回数を決める運行回数決定モデルによりシステムを構成した。

バス路線網計画問題は、通常の最適交通網構成問題をさらに複雑にしたものといえる。そのため、厳密な最適解を求められないのは明らかである。そこで、ここでは実用性を重視したシステムを開発した。ケーススタディとして尼崎市内のバス路線網を再編成する問題に適用した結果、実際の問題に十分適用可能であることを示せた。システムの適用にあたっては、モデルの前提条件と地域の特性を考慮することが必要であるが、これによりバス路線網の決定を合理的、容易に行うことができると考える。

6.7　結　　語

この章では、その実用性と応用の可能性を示すために、最適交通網構成手法をいくつかの交通網計画問題へ適用した。すなわち、大量輸送機関を組み合わせる地域交通網計画問題、複数の目的関数を扱う道路網計画問題、道路網の対災害信頼性の最適化問題、配分対象道路網を作成する問題、バス路線系統の組み合わせを求める問題を解くバス路線網計画問題への適用を行った。それぞれの問題を最適交通網構成問題として定式化し、それに適する解法を提案した。

地域交通網計画問題は、鉄道、新交通システム、バスからなる地域の交通網を決めるものである。この問題の特徴は次のとおりである。

① 3種類の大量輸送機関を対象とし、最大ネットワークにおいては同一のノード間に複数の検討リンクが存在する場合がある。

② 鉄道リンクは都心側のリンクがなければネットワークに含めず、鉄道リンクが孤立しないようにする。

これらの点を考慮し、局所最適化を行う近似解法を修正して用いた。そして、ケーススタディとして北摂・北神地域の交通網計画へ適用した。最適交通網構成問題は問題を単純化しているため、定式化に組み込めない評価項目や制約条件がある。ここでは、得られた解を検討し、必要な修正を加えて再計算することで、それに対処した。その結果、妥当な交通網計画案が得られた。この過程は、最適交通網構成手法の現実的な使用法を示したものといえる。

道路網計画問題は、総走行時間、総走行距離、騒音の影響、排出ガスの影響を最小化するものである。この問題の特徴は、次のとおりである。

① 多目的最適交通網構成問題である。

② 交通配分に等時間配分を用いる。

解法には、あらかじめ非劣解を列挙しておく方法を用いた。そして、等時間配分は限られた計算過程でだけ使用し、計算時間の増加をおさえた。ケーススタディとしては、神戸市の道路網への適用を行った。その結果、複数の目的関数を考慮し、等時間配分を行うという複雑な最適交通網構成問題を実用的な水準で解けることを示せた。

道路網の災害に対する信頼性を最適化する問題は、これまで考えられていなかったものである。道路網の対災害信頼性を評価するために、リンクの強さを表わすリンクレベルや災害強度レベル、災害パターンの概念を導入した。そして、災害時の総所要時間を最小化する問題と災害時の到達トリップ数を最大化する問題を定式化した。決定変数がリンクレベルなので、それにあわせて基本問題に対する解法を修正した。

ここで示した問題は、実際の自然災害を分析して導いたものではない。したがって、問題の前提が必ずしも現実的でないかもしれない。また提案した解法は厳密解法であり、実用性に欠ける。しかし、最適交通網構成問題の考え方から、災害時のネットワークとしての信頼性を考慮して道路網計画を行う新しい方法が生まれたことに意義がある。今後は、災害による道路の破壊や災害時に要求される道路網の機能などを検討するとともに、実用的な解法をつくることが必要である。

配分対象道路網作成問題は、交通配分の目的を損わないで計算量を減らせる簡略化した配分対象道路網を求めるものである。交通配分を行うときには道路網の設定作業が必ず必要である。配分対象道路網の作成が交通網計画に類する問題であるとは考えられていなかった。ここでは問題を整理し、最適交通網構成手法を用いて配分対象道路網を求める方法を提案した。この問題では、大規模な道路網を扱うことが前提である。そこで、解法には大規模問題への適用が可能な近似解法を用いた。計算例においては、求めた配分対象道路網での配分結果から、詳細な道路網での着目リンクの交通量を精度よく推定できた。簡単な例題に対してではあるが、この手法が有効であることが示されたのである。

バス路線網計画問題は、バス道路網と路線系統、運行回数を決めるものである。このうち、バスを運行する道路網を決める問題と路線系統の最適な組み合わせを求める問題に最適交通網構成手法を適用した。また、路線列挙問題は最適経路探索問題であり、最適交通網構成問題であるともいえる。路線網決定問題は、次の特徴をもつ。

① 路線系統の組み合わせを最適化する。

② 乗り換えを考慮した最短経路探索を行う。

このために、解を得るのに必要な計算量は大きい。そこで、解法は backward 法を用いた。ケーススタディでは、尼崎市のバス路線網を再編成する問題を解いた。そして、バス道路網の限定から運行回数の決定までを一貫して行うシステムの実用性が実証された。このバス路線網構成システムの主要部分には最適交通網構成手法が用いられており、これによりバス路線網の決定を合理的、容易に行えるようになったのである。

このように、最適交通網構成手法は多様な交通網計画問題へ適用できる。適用例のなかには、まだ実用段階でないものもある。しかし、最適交通網構成問題の考え方に基づき、新しい交通網

計画問題を示すことができた。最適交通網構成手法の実用性と応用の可能性を示すという、この章の目的は達せられた。

それぞれの問題においては、最適交通網構成問題として、どう定式化するか、解法はどのようにすればよいかを述べた。これは、最適交通網構成手法を交通網計画の基礎的な手法として用いるときに役立つであろう。個々の交通網計画問題を深く掘り下げることが、今後の課題である。

参考文献

1） 枝村俊郎・森津秀夫：最適ネットワーク問題の地域交通網計画への応用 ，交通工学，第12巻，4号，pp.23～31，昭和52年７月．

2） 東京都市群交通計画委員会：東京50km圏総合交通体系調査報告書 ，技術部会資料，新交通システム編，1972．

3） 運輸経済研究センター：神奈川県内新交通輸送体系確立に関する調査報告書，昭和48年３月．

4） 枝村俊郎・田中　稔・森津秀夫・藤谷恵一：環境要因を考慮した最適道路網計画システムと神戸市への適用について ，建設工学研究所報告，第19号，pp.187～206，昭和52年５月．

5） 枝村俊郎・森津秀夫・山本卓司：複数目標を考慮した最適ネットワーク問題，建設工学研究所報告，第23号，pp.167～176，昭和56年11月．

6） 杜若善彦・森津秀夫：複数目標下の最適道路網計画モデル，土木学会第37回年次学術講演会講演概要集，第４部，pp.397～398，昭和57年10月．

7） 大根田洋祐・渡辺　隆・森地　茂：環境要因を考慮した街路網計画手法について，土木学会第31回年次学術講演会講演概要集，第４部，pp.148，昭和51年10月．

8） 吉川和広・小林潔司・川合紀章：目標計画法による広域的な幹線道路網整備計画問題に関するシステム分析，土木学会第34回年次学術講演会講演概要集，第４部，pp.49～50，昭和54年10月．

9） 金安公造・金泉　昭：交通公害，技術書院，昭和51年10月．

10） 加来治郎・山下充康：市街地における騒音予測のひとつのこころみ，日本音響学会講演論文集，pp.11～12，1978年５月．

11） 自動車排出ガスの調査委員会：自動車排出ガスの調査研究報告書，1975．

12） 志水清孝：多目的と競争の理論，共立出版，昭和57年６月．

13） Steenbrink, P. A. : Optimization of Transport Networks, John Wiley & Sons, 1974.

14） Edamura, T. , H. Moritsu, M. Doi and S. Nakagawa : Optimal allocation of reliability on a transportation network, Memoirs of the Faculty of Engineering, Kobe University, №28, pp.133～157, March, 1982.

15) Hoang, H. H. : A computational approach to the selection of an optimal network, Management Science, Vol. 15, No. 5 , pp.488～498, January, 1973.

16） Jensen, P. A. and M. Bellmore : An algorithm to determine the reliability of a complex system, IEEE Transactions on Reliability, Vol. R－18, pp.169～174, Novemver, 1969.

17） 枝村俊郎・森津秀夫・木下暢男・樋口和夫：配分対象道路網作成の自動化，第３回土木計画学研究発表会講演集，pp.341～349，昭和56年１月．

18） Chan, Y－P., K. G. Follansbee, M. L. Manheim and J. R. Mumford : Aggregation in Transport Networks : An Application of Hierarchical Structure, Research Report R68－47, M. I. T. , June, 1968.

19） 日本オペレーションズ・リサーチ学会：ネットワーク構造を有するオペレーションズ・リサーチ問

題の電算機処理に関する基礎研究，報文シリーズ・T－73－１，1973年３月．
20)　日本道路公団経済調査室：新手法による高速道路交通量の推計，報文シリーズT－73－２，昭和48年２月．
21)　枝村俊郎・森津秀夫・松田　宏・土井元治：最適バス路線網構成システム，土木学会論文報告集，第300号，pp.95～107，1980年８月．

第7章　結　　論

本研究の目的は、最適交通網構成手法を実用的な水準に近づけるための基礎的な考察を行うことであった。そのために、最適解を効率よく求められる解法の作成と、実用的な近似解法の作成、現実的な交通網計画問題への応用を行った。

まず第2章では、最適交通網構成問題に関する基礎的な考察を行った。つぎに、第3章と第4章で、基本的な問題を対象に厳密解法のアルゴリズムと近似解法を提案した。第5章では、現実的な交通網計画に適用するために、基本的な問題を拡張した。すなわち、制約条件が複数の問題、目的関数が複数の問題、交通網を段階的に建設する問題、それに需要交通量が不確実な場合の問題を取り上げ、それぞれの解法を提案した。そして、第6章では、地域交通網計画問題、多目的を扱う道路網計画問題、道路網の対災害信頼性の最適化問題、配分対象道路網作成問題、およびバス路線網計画問題へ最適交通網構成手法を適用し、現実的な交通網計画の定式化法と解法の拡張方法を示した。

以下に、各章で得られた成果について述べる。

第2章では、この研究の対象である最適交通網構成問題を明確にするために基礎的な考察を行った。すなわち、交通網計画への最適化手法の導入の必要性を明らかにし、最適交通網構成問題の定式化と解法の概要を示した。そして、基本的な問題の拡張にどのようなものがあるかを述べた。

第2章での成果は、次のようにまとめられる。

① 交通網計画において、代替案を比較する従来の手法の持つ欠点は最適化手法の導入により除けることを示し、最適交通網構成手法の必要性を明らかにした。

② 最適交通網構成問題の解法に関し、使用できる数理計画的手法と問題点を示し、解法の研究の課題を明らかにした。

③ 実際の交通網計画に適用しようとするとき、解法の開発に用いた基本問題をどのように拡張しなければならないかを示し、最適交通網構成手法の応用を図る際の課題を明らかにした。

第3章では、最適交通網構成問題の基本問題を対象に、厳密解法のアルゴリズムを改良した。まず、従来のアルゴリズムの検討の結果として、分枝後退法が分枝限定法よりも扱いやすいこと、アルゴリズムの改良には一度に複数の分枝変数の値を固定する分枝操作の採用と、ネットワークが非連結になる解を作らない分枝方法の採用とが考えられることを明らかにした。そして、これらの考察をもとに、実行可能化分枝と連結網化分枝の2種類の分枝方法を提案した。さらに、例題を解いて、提案した分枝方法を使った場合と使わない場合とを比較し、計算時間の短縮の効果を確認した。

第3章での成果は、次のようにまとめられる。

① 分枝方法には分枝限定法よりも分枝後退法を採用する方が、近似解法への応用を図るためにも好ましいことを明確にした。

② 一度に複数の分枝変数の値を固定する実行可能化分枝を提案し、これにより、計算時間を

短縮できた。

③ 非連結網である解を作らない連結網化分枝を提案した。これにより、従来は計算時間が極めて長かったネットワーク長の制約値の小さい場合でも、短い計算時間で解が得られるようになった。

第4章では、最適交通網構成問題の基本問題を対象に、新しい近似解法を提案した。まず、従来の近似解法の検討の結果から、近似解法の作成に際しては厳密解法での計算例の考察を生かし、厳密解法の応用を図るのがよいことと、大規模な問題を解ける近似解法を作る必要があることを指摘した。そして、厳密解法を応用したものとして、リンクの段階的削減を行う近似解法と局所最適化を行う近似解法を提案した。さらに、簡単なリンク評価値を使い、forward 法と backward 法の手順を用いる近似解法、および、これと厳密解法の応用とを組み合わせた近似解法を提案した。これらの近似解法が優れていることは、例題を解いて解の精度と計算時間を従来の近似解法と比較して確かめた。

第4章での成果は、次のようにまとめられる。

① 厳密解法での計算例の考察から近似解法を改良する方向を導いた。

② 厳密解法を応用し、リンクの段階的削減を行う近似解法と局所最適化を行う近似解法を提案した。これにより、従来よりも短い計算時間で精度の高い解が求められるようになった。また、パラメータの値を変えて、解の精度や計算時間を制御することが可能になった。

③ 最大ネットワークでの経路探索で求めた値から計算したリンク評価値を使う簡易 forward 法と簡易 backward 法を提案した。これにより、従来よりも大規模な問題を解くことが可能になった。

④ 簡易 forward 法、簡易 backward 法と段階的削減法、局所最適化法とを組み合わせた近似解法も提案した。これにより、さらに短い計算時間で精度の高い解を求められるようになった。

第5章では、最適交通網構成問題を基本問題から拡張したとき、どのように解法を修正しなければならないかを考察した。すなわち、拡張の基本的なものとして、複数の制約条件を持つ場合、複数の目的関数を持つ場合、段階的に交通網を建設する場合、不確実な需要交通量を用いる場合を取り上げ、各問題の解法を提案した。

第5章での成果は、次のようにまとめられる。

① 複数の制約条件を持つ場合を制約条件の組み合わせで4種類に分け、それぞれのときの解法を検討し、基本問題と類似の解法が使える問題を明確にした。そして、制約式が多次元ナップザック問題と同じ形の問題に対し、多次元ナップザック問題の近似解法を応用した厳密解法と近似解法を提案した。

② 多目的最適交通網構成問題については、一般の多目的計画問題に対する手法のうち、目的関数のスカラ化手法が適用できることを明らかにした。そして、最適交通網構成問題の場合に重要な計算時間の問題を解決した計算手順を提案した。これにより、多目的最適交通網構

成問題を実用的な水準で扱うことが可能になった。

③ 交通網を段階的に建設する問題では、リンクの建設時期が決定変数となるため、解の探索木が基本問題とは異なるが、これに適する解法を提案した。

④ 需要交通量が不確実なとき、どのような問題で解の決定に影響があるかを明らかにした。そして、総走行距離とリンクの超過交通量のペナルティーの和を最小化する問題の解法を提案し、需要交通量の不確実さを考慮しないで交通網を決めると、損失を被る可能性があることを計算例で示せた。

第6章では、最適交通網構成手法をいくつかの交通網計画問題に適用し、その実用性と応用の可能性を示した。すなわち、地域交通網計画問題、多目的を扱う道路網計画問題、道路網の対災害信頼性の最適化問題、配分対象道路網作成問題、さらにバス路線網計画問題への適用例を示し、これらによって、現実的な交通網計画の定式化法と解法の拡張方法を明らかにした。

第6章での成果は、次のようにまとめられる。

① 地域交通網計画問題に対しては、3種類の大量輸送機関の組み合わせを扱える解法を提案した。そして、ケーススタディにおいては必要な修正を加えて再計算し、定式化に組み込めない評価項目や制約条件を考慮する過程をとり、最適交通網構成手法の現実的な使用法を示せた。

② 総走行時間、総走行距離、騒音の影響、排出ガスの影響を最小化する道路網計画問題を多目的最適交通網構成問題として解き、この手法の実用性を明らかにした。多目的の複雑な問題ではあるが、提案した解法では交通配分には等時間配分を用い、より現実に近い状態を考えることが可能になった。

③ 道路網の災害に対する信頼性を最適化する問題では、道路網の対災害信頼性を評価するために、リンクレベルや災害強度レベル、災害パターンの概念を導入した。そして、信頼性を最適化する方法として、災害時の総所要時間を最小化する問題と災害時の到達トリップ数を最大化する問題を明らかにし、その解法を提案した。これにより、最適交通網構成手法を使って、従来は考えられていなかった災害時のネットワークとしての信頼性を考慮した道路網計画が可能になった。

④ 配分対象道路網作成問題に関しては、これが交通網計画問題に類するものであることを示し、大規模問題への適用が可能な近似解法を使った解法を提案した。そして、計算例では求めた配分対象道路網での配分結果から、詳細な道路網でのリンク交通量を精度よく推定でき、実用性を確認できた。

⑤ バス路線網計画問題では、バスを運行する道路網を決める問題と、路線系統の最適な組み合わせを求める問題に対する最適交通網構成手法による解法を提案した。そして、これを組み込み、バス道路網から運行回数までを一貫して定められるシステムを提案し、ケーススタディによって実用性を示せた。これにより、バス路線網の決定を合理的、容易に行うことが可能になった。